

C++ Programming Questions and Answers – Basics

1. Which of the following is the correct syntax of including a user defined header files in C++?

- a) `#include <userdefined.h>`
- b) `#include <userdefined>`
- c) `#include "userdefined"`
- d) `#include [userdefined]`

View Answer

Answer: c

Explanation: C++ uses double quotes to include a user-defined header file. The correct syntax of including user-defined is `#include "userdefinedname"`.

2. Which of the following is a correct identifier in C++?

- a) 7var_name
- b) 7VARNAME
- c) VAR_1234
- d) \$var_name

View Answer

Answer: c

Explanation: The rules for writing an identifier is as follows:

- i) may contain lowercase/uppercase letters, digits or underscore(_) only
 - ii) should start with a non-digit character
 - iii) should not contain any special characters like @, \$, etc.
-

3. Which of the following is called address operator?

- a) *
- b) &
- c) _
- d) %

View Answer

Answer: b

Explanation: & operator is called address operator and is used to access the address of a variable.

4. Which of the following is used for comments in C++?

- a) `// comment`
- b) `/* comment */`
- c) both `// comment` or `/* comment */`
- d) `// comment */`

View Answer

Answer: c

Explanation: Both the ways are used for commenting in C++ programming. `//` is used for single line comments and `/* ... */` is used for multiple line comments.

5. What are the actual parameters in C++?

- a) Parameters with which functions are called
- b) Parameters which are used in the definition of a function
- c) Variables other than passed parameters in a function
- d) Variables that are never used in the function

View Answer

Answer: a

Explanation: Actual parameters are those using which a function call is made i.e. which are actually passed in a function when that function is called.

6. What are the formal parameters in C++?

- a) Parameters with which functions are called
- b) Parameters which are used in the definition of the function
- c) Variables other than passed parameters in a function
- d) Variables that are never used in the function

View Answer

Answer: b

Explanation: Formal parameters are those which are used in the definition of a function. They are the parameters that represent the actual parameters passed and they are the one which is used inside the function.

7. Which function is used to read a single character from the console in C++?

- a) `cin.get(ch)`
- b) `getline(ch)`
- c) `read(ch)`
- d) `scanf(ch)`

View Answer

Answer: a

Explanation: C++ provides `cin.get()` function to read a single character from console whereas others are used to read either a single or multiple characters.

8. Which function is used to write a single character to console in C++?

- a) `cout.put(ch)`
- b) `cout.putline(ch)`
- c) `write(ch)`
- d) `printf(ch)`

View Answer

Answer: a

Explanation: C++ provides `cout.put()` function to write a single character to console whereas others are used to write either a single or multiple characters.

9. What are the escape sequences?

- a) Set of characters that convey special meaning in a program
- b) Set of characters that whose use are avoided in C++ programs
- c) Set of characters that are used in the name of the main function of the program
- d) Set of characters that are avoided in `cout` statements

View Answer

Answer: a

Explanation: Escape sequence is a set of characters that convey a special meaning to the program. They are used to convey a meaning which cannot be conveyed directly.

10. Which of the following escape sequence represents carriage return?

- a) `\r`
- b) `\n`
- c) `\nr`
- d) `\c`

View Answer

Answer: a

Explanation: `\r` is used to represent carriage return which means move the cursor to the beginning of the next line.

11. Which of the following escape sequence represents tab?

- a) \t
- b) \tr
- c) \b
- d) \a

View Answer

Answer: a

Explanation: \t is used to represent tab which means a set of blank spaces in a line.

12. Who created C++?

- a) Bjarne Stroustrup
- b) Dennis Ritchie
- c) Ken Thompson
- d) Brian Kernighan

View Answer

Answer: a

Explanation: Bjarne Stroustrup is the original creator of C++ during 1979 at AT&T Bell Labs.

13. Which of the following is called insertion/put to operator?

- a) <<
- b) >>
- c) >
- d) <

View Answer

Answer: a

Explanation: << operator is called insertion or put to operator i.e. insert/put things to console/files.

14. Which of the following is called insertion/put to operator?

- a) <<
- b) >>
- c) >
- d) <

View Answer

Answer: b

Explanation: >> operator is called extraction or get from operator i.e. extract/get things from console/files.

15. A language which has the capability to generate new data types are called _____

- a) Extensible
- b) Overloaded
- c) Encapsulated
- d) Reprehensible

View Answer

Answer: a

Explanation: Languages that can produce/generate new data types are called extensible languages as they have the ability to handle new data types.

C++ Programming Questions and Answers – OOPs Concept – 1

1. Wrapping data and its related functionality into a single entity is known as _____

- a) Abstraction
- b) Encapsulation
- c) Polymorphism
- d) Modularity

View Answer

Answer: b

Explanation: In OOPs, the property of enclosing data and its related functions into a single entity (in C++ we call them classes) is called encapsulation.

2. How structures and classes in C++ differ?

- a) Classes follow OOP concepts whereas structure does not
- b) In Structures, members are private by default whereas in Classes they are public by default
- c) Structures by default hide every member
- d) Classes and Structures are the same

View Answer

Answer: a

Explanation: Structures do not follow OOPs concepts as structures do not keep fields and methods together with their interactions whereas classes do.

3. What does polymorphism in OOPs mean?

- a) Concept of allowing overriding of functions
- b) Concept of hiding data
- c) Concept of keeping things in different modules/files
- d) Concept of wrapping things into a single unit

View Answer

Answer: a

Explanation: In OOPs, Polymorphism is the concept of allowing a user to override functions either by changing the types or number of parameters passed.

4. Which concept allows you to reuse the written code?

- a) Encapsulation
- b) Abstraction
- c) Inheritance
- d) Polymorphism

View Answer

Answer: c

Explanation: Inheritance allows you to reuse your already written code by inheriting the properties of written code into other parts of the code, hence allowing you to reuse the already written code.

View Answer

Answer: c

Explanation: Polymorphism means overriding the same function by changing types or number of arguments. So we have only two options which have the same function names, but as one can observe that in one option types, name and number of parameters are all the same which will lead to an error. Hence that is wrong so the option having same name and different types or number of parameters is correct.

6. Which of the following shows multiple inheritances?

- a) A->B->C
- b) A->B; A->C

c) A,B->C

d) B->A

View Answer

Answer: c

Explanation: In multiple inheritance, a single class is inherited from two classes. So in A,B->C, Class C is inherited from A and B, whereas in A->B->C, C from B and B from A called simple inheritance, in A->B; A->C, B and C are inherited from A which is called hierarchical inheritance.

7. How access specifiers in Class helps in Abstraction?

a) They does not helps in any way

b) They allows us to show only required things to outer world

c) They help in keeping things together

d) Abstraction concept is not used in classes

View Answer

Answer: b

Explanation: Abstraction is the concept of hiding things from the outer world and showing only the required things to the world, which is where access specifiers private, protected and public helps in keeping our knowledge hidden from the world.

8. C++ is _____

a) procedural programming language

b) object oriented programming language

c) functional programming language

d) both procedural and object oriented programming language

View Answer

Answer: d

Explanation: C++ supports both procedural(step by step instruction) and object oriented programming(using concept of classes and objects).

9. What does modularity mean?

a) Hiding part of program

b) Subdividing program into small independent parts

c) Overriding parts of program

d) Wrapping things into single unit

View Answer

Answer: b

Explanation: Modularity means dividing a program into independent sub programs so that it can be invoked from other parts of the same program or any other program.

a) Abstraction

b) Encapsulation

c) Inheritance

d) Polymorphism

View Answer

Answer: d

EXplanation: As i and j members are private i.e. they are hidden from outer world therefore we have used the concept of abstraction. Next data members and there related functions are put together into single class therefore encapsulation is used. Also as class B is derived from A therefore Inheritance concept is used. But as no function is overloaded in any of the classes therefore, the concept of polymorphism is missing here.

C++ Programming Questions and Answers – OOPs Concept – 2

1. Which of the following class allows to declare only one object of it?

- a) Abstract class
- b) Virtual class
- c) Singleton class
- d) Friend class

View Answer

Answer: c

Explanation: Singleton class allows the programmer to declare only one object of it, If one tries to declare more than one object the program results into error.

2. Which of the following is not a type of Constructor?

- a) Friend constructor
- b) Copy constructor
- c) Default constructor
- d) Parameterized constructor

View Answer

Answer: a

Explanation: Friend function is not a constructor whereas others are a type of constructor used for object initialization.

3. Which of the following is correct?

- a) Base class pointer object cannot point to a derived class object
- b) Derived class pointer object cannot point to a base class object
- c) A derived class cannot have pointer objects
- d) A base class cannot have pointer objects

View Answer

Answer: b

Explanation: C++ does not allow a derived class pointer to point a base class pointer whereas Base class can point to a derived class object. Both base class and derived class can have pointer objects.

4. Out of the following, which is not a member of the class?

- a) Static function
- b) Friend function
- c) Constant function
- d) Virtual function

View Answer

Answer: b

Explanation: Friend function is not a member of the class. They are given the same access rights as the class member function have but they are not actual members of the class.

5. What is the other name used for functions inside a class?

- a) Member variables
- b) Member functions
- c) Class functions
- d) Class variables

View Answer

Answer: b

Explanation: Functions of a class are also known as member functions of a class.

6. Which of the following cannot be a friend?

- a) Function
- b) Class
- c) Object
- d) Operator function

View Answer

Answer: c

Explanation: Objects of any class cannot be made a friend of any other or same class whereas functions, classes and operator functions can be made a friend.

7. Why references are different from pointers?

- a) A reference cannot be made null
- b) A reference cannot be changed once initialized
- c) No extra operator is needed for dereferencing of a reference
- d) All of the mentioned

View Answer

Answer: d

Explanation: References cannot be made null whereas a pointer can be. References cannot be changed whereas pointers can be modified. Pointers need * operator to dereference the value present inside it whereas reference does not need an operator for dereferencing.

8. Which of the following provides a programmer with the facility of using object of a class inside other classes?

- a) Inheritance
- b) Composition
- c) Abstraction
- d) Encapsulation

View Answer

Answer: b

Explanation: The concept of using objects of one class into another class is known as Composition.

9. How many types of polymorphism are there in C++?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: b

Explanation: There are two types of polymorphism in C++ namely run-time and compile-time polymorphisms.

10. How run-time polymorphisms are implemented in C++?

- a) Using Inheritance
- b) Using Virtual functions
- c) Using Templates
- d) Using Inheritance and Virtual functions

View Answer

Answer: d

Explanation: Run-time polymorphism is implemented using Inheritance and virtual in which object decides which function to call.

11. How compile-time polymorphisms are implemented in C++?

- a) Using Inheritance
- b) Using Virtual functions
- c) Using Templates
- d) Using Inheritance and Virtual functions

View Answer

Answer: c

Explanation: Compile-time polymorphism is implemented using templates in which the types(which can be checked during compile-time) are used decides which function to be called.

12. Which of the following is an abstract data type?

- a) int
- b) float
- c) class
- d) string

View Answer

Answer: c

Explanation: Class is used as an abstract data type as it can be used to give implementation independent view whereas no other data type can be used to provide this.

13. Which concept means the addition of new components to a program as it runs?

- a) Data hiding
- b) Dynamic binding
- c) Dynamic loading
- d) Dynamic typing

View Answer

Answer: c

Explanation: Dynamic loading is the concept of adding new components to a program as it runs.

14. Which of the following explains the overloading of functions?

- a) Virtual polymorphism
- b) Transient polymorphism
- c) Ad-hoc polymorphism
- s) Pseudo polymorphism

View Answer

Answer: c

Explanation: Ad-hoc polymorphism is a type of polymorphism in which a function denotes heterogeneous implementation depending upon the types of argument.

15. Which of the following approach is used by C++?

- a) Top-down
- b) Bottom-up
- c) Left-right
- d) Right-left

View Answer

Answer: b

Explanation: C++ is an object-oriented language and OOL uses a bottom-up approach to solve/view a problem.

C++ Programming Questions and Answers – OOPs Concept – 3

1. Which operator is overloaded for a cout object?

- a) >>
- b) <<
- c) <
- d) >

View Answer

Answer: b

Explanation: cout in C++ uses << operator to print anything so << operator is overloaded for a cout object.

2. Which of the following cannot be used with the virtual keyword?

- a) Class
- b) Member functions
- c) Constructors
- d) Destructors

View Answer

Answer: c

Explanation: Virtual keyword cannot be used with constructors as constructors are defined to initialize an object of particular class hence no other class needs constructor of other class.

3. Which concept is used to implement late binding?

- a) Virtual functions
- b) Operator functions
- c) Constant functions
- d) Static functions

View Answer

Answer: a

Explanation: Virtual functions are used to implement the concept of late binding i.e. binding actual functions to their calls.

4. Which of the following is correct?

- a) C++ allows static type checking
- b) C++ allows dynamic type checking.
- c) C++ allows static member function to be of type const.
- d) C++ allows both static and dynamic type checking

View Answer

Answer: d

Explanation: C++ allows both static and dynamic type checking i.e. types are checked by the compiler.

5. Which of the following supports the concept that reusability is a desirable feature of a language?

- a) It reduces the testing time
- b) It reduces maintenance cost
- c) It decreases the compilation time
- d) It reduced both testing and maintenance time

View Answer

Answer: d

Explanation: As we will be using the existing code therefore we don't need to check the code again and again so testing and maintenance time decreases but the compiler time may increase or remains same because though we are reusing the code but every part needs to be compiled and extra include statement needs to be executed therefore compilation time may remain same or increases.

6. Which of the following is a static polymorphism mechanism?

- a) Function overloading
- b) Operator overloading
- c) Templates
- d) All of the mentioned

View Answer

Answer: d

Explanation: All the options mentioned above uses static polymorphism mechanism. As the conflicts in all these types of functions are resolved during compile-time.

7. Which of the following is true?

- I) All operators in C++ can be overloaded.
- II) The basic meaning of an operator can be changed.

- a) I only
- b) II only
- c) Both I and II
- d) Neither I nor II

View Answer

Answer: d

Explanation: Both statements are false because all the operators of C++ cannot be overloaded and the basic meaning of an operator cannot be changed, we can only give new meaning to an operator.

8. Which of the following is not a type of inheritance?

- a) Multiple
- b) Multilevel
- c) Distributive
- d) Hierarchical

View Answer

Answer: c

Explanation: Distributive is not a type of inheritance whereas others are a type of inheritance having their own meaning.

9. What happens if a class does not have a name?

- a) It will not have a constructor
- b) It will not have a destructor
- c) It is not allowed
- d) It will neither have a constructor or destructor

View Answer

Answer: b

Explanation: A class without a name will not have a destructor. The object is made so constructor is required but the destructor is not. Check the code below:

```
#include <iostream>
using namespace std;
class
{
    public:
        void func()
        {
            cout<<"Hello world";
        }
}a;
int main(int argc, char const *argv[])
{
    a.func();
    return 0;
}
```

10. Which of the following statement is true?

- I) In Procedural programming languages, all function calls are resolved at compile-time
- II) In Object Oriented programming languages, all function calls are resolved at compile-time
- a) I only
- b) II only
- c) Both I and II
- d) Neither I nor II

View Answer

Answer: a

Explanation: In Procedural programming like C we don't have the concept of polymorphism, therefore, all the function calls are resolved at the compile-time but in case of OOP languages due to polymorphism concept all function calls are not resolved at compile-time.

11. Which members are inherited but are not accessible in any case?

- a) Private
- b) Public
- c) Protected
- d) Both private and protected

View Answer

Answer: a

Explanation: Private members of a class are inherited to the child class but are not accessible from the child class.

12. Which of the following is correct?

- a) Friend functions can access public members of a class
- b) Friend functions can access protected members of a class
- c) Friend functions can access private members of a class
- d) All of the mentioned

View Answer

Answer: d

Explanation: Friend functions can access any member of a class without caring about the type of member i.e. without caring whether it is private, protected or public.

13. Which of the following is correct in C++?

- a) Classes cannot have protected data members
- b) Structures can have member functions
- c) Class members are public by default
- d) Structure members are private by default

View Answer

Answer: b

Explanation: Though C does not allow member functions in structures but C++ allows structures to have member functions. Members of structures are public by default and those of classes are private by default. Classes can have protected data members.

14. Which of the following is used to make an abstract class?

- a) By using virtual keyword in front of a class declaration
- b) By using an abstract keyword in front of a class declaration
- c) By declaring a virtual function in a class
- d) By declaring a pure virtual function in a class

View Answer

Answer: d

Explanation: Abstract class should have at least one pure virtual function. Therefore to declare an abstract class one should declare a pure virtual function in a class.

15. Which of the following is correct?

- a) A class is an instance of its objects
- b) An object is an instance of its class
- c) A class is an instance of the data type that the class have
- d) An object is an instance of the data type of the class

View Answer

Answer: b

Explanation: An object is an instance of a class i.e. an object represents a class i.e. what class has(data members) and what it can do(member functions).

ayzom.com

C++ Programming Questions and Answers – OOPs Concept – 4

1. Which of the following is correct about new and malloc?

- a) Both are available in C
- b) Pointer object initialization of a class with both new and malloc calls the constructor of that class
- c) Pointer object initialization of a class using new involves constructor call whereas using malloc does not involve constructor call
- d) Pointer object initialization of a class using malloc involves constructor call whereas using new does not involve constructor call

View Answer

Answer: c

Explanation: Object initialization using new keyword involves constructor call whereas malloc does not involve constructor call. That's why new is explicitly added in C++. Also, malloc is used to assign memory to any pointer hence it assigns memory equals to the size of the class however new keyword involves initialization also hence calls the constructor of that class.

2. What is virtual inheritance?

- a) C++ technique to avoid multiple copies of the base class into children/derived class
- b) C++ technique to avoid multiple inheritances of classes
- c) C++ technique to enhance multiple inheritance
- d) C++ technique to ensure that a private member of the base class can be accessed somehow

View Answer

Answer: a

Explanation: Virtual inheritance is a C++ technique with which it ensures that a derived class contains only one copy of the base class's variables. Refer Wikipedia for more info.

3. What is the difference between delete and delete[] in C++?

- a) delete is used to delete normal objects whereas delete[] is used to pointer objects
- b) delete is a keyword whereas delete[] is an identifier
- c) delete is used to delete single object whereas delete[] is used to multiple(array/pointer of) objects
- d) delete is syntactically correct but delete[] is wrong and hence will give an error if used in any case

View Answer

Answer: c

Explanation: delete is used to delete a single object initiated using new keyword whereas delete[] is used to delete a group of objects initiated with the new operator.

- a) "Constructor called" five times and then "Destructor called" five times
- b) "Constructor called" five times and then "Destructor called" once
- c) Error
- d) Segmentation fault

View Answer

Answer: d

Explanation: The program will result in segmentation fault as we are trying to delete only one pointer variable and leaving other variables as it is which will result in segmentation fault i.e. improper handling of memory.

- a) "Constructor called" five times and then "Destructor called" five times
- b) "Constructor called" five times and then "Destructor called" once
- c) Error
- d) Segmentation fault

View Answer

Answer: a

Explanation: In the above program we have first initiated five-pointer variables using new keyword hence five times constructor will be called after that as we using delete[] (used for deleting multiple objects) to delete variables hence all the five objects created will be destroyed and

hence five times destructor will be called.

View Answer

Answer: a

Explanation: As we are storing a derived class object into base class pointer therefore when the object is destroyed the program has not called the Derived class destructor which shows that the object is not destroyed therefore the program may give unusual behaviour.

View Answer

Answer: b

Explanation: In this case, we have made the destructor of base class virtual which will ensure that any derived class object which is pointed by a base class pointer object on deletion should call both base and derived class destructor.

8. What is the correct syntax of declaring array of pointers of integers of size 10 in C++?

- a) `int arr = new int[10];`
- b) `int **arr = new int*[10];`
- c) `int *arr = new int[10];`
- d) `int *arr = new int*[10];`

View Answer

Answer: b

Explanation: As we have to declare an array of pointers of integers therefore we need double pointer array in which each element is collection pointers to integers. Therefore the correct syntax is `int **arr = new int*[10];`

9. Which of the following is correct about new and malloc?

- i) new is an operator whereas malloc is a function
 - ii) new calls constructor malloc does not
 - iii) new returns required pointer whereas malloc returns void pointer and needs to be typecast
- a) i and ii
 - b) ii and iii
 - c) i and iii
 - d) i, ii and iii

View Answer

Answer: d

Explanation: All the statements about the new and malloc are correct. new is an operator whereas malloc() is a function. The constructor is called when new is used and new returns required type memory pointer.

- a) 5
- b) Garbage value
- c) Compile-time error
- d) Run-time error

View Answer

Answer: c

Explanation: As Test() constructor is private member of the class therefore cannot be accessed from the outside world therefore the program gives error.

- a) The program compiled successfully but throws an error during run-time
- b) The program gives a compile-time error
- c) The program is not semantically correct
- d) The program is compiled and executed successfully

View Answer

Answer: d

Explanation: The above statement is syntactically and semantically correct as C++ allows the programmer to delete a NULL pointer, therefore, the program is compiled and executed successfully.

- a) Undefined behaviour
- b) Syntactically incorrect
- c) Semantically incorrect
- d) The program runs perfectly

View Answer

Answer: a

Explanation: Deleting a pointer twice in a program may lead to run-time error or may run perfectly. It depends on the compiler how it handles the situation so the program may compile and run successfully but actually the program should give a run-time error(segmentation fault) as you are trying to access the unauthorized memory of the system.

ayzom.com

C++ Programming Questions and Answers – C++ vs C

3. What happens if the below line is executed in C and C++?

```
int *p = malloc(10);
```

- a) Error in both C and C++
- b) Warning in both C and C++
- c) Error in C++ but Warning in C
- d) Error in C but Warning in C++

View Answer

Answer: c

Explanation: In C++ all the functions should be declared before it is called otherwise the C++ compiler will give an error but in case of C the compiler just gives a warning and the program can be executed.

3. What happens if the below line is executed in C and C++?

```
int *p = malloc(10);
```

- a) Error in both C and C++
- b) Warning in both C and C++
- c) Error in C but Warning in C++
- d) Error in C++ but Warning in C

View Answer

Answer: d

Explanation: C++ is strict on the use of types of variables hence when the programmer tries to assign const int to a normal pointer the program gives error whereas C is not strict on types therefore it gives warning only.

3. What happens if the below line is executed in C and C++?

```
int *p = malloc(10);
```

- a) Error in both C and C++
- b) Warning in both C and C++
- c) Error in C++ and successful execution in C
- d) Error in C and successful execution in C++

View Answer

Answer: c

Explanation: C++ is strict in type check but C is not and as malloc returns a void* which we are trying to assign to an int*, therefore, the C++ compiler gives error whereas C compiler executes the program successfully.

3. What happens if the below line is executed in C and C++?

```
int *p = malloc(10);
```

- a) Error in both C and C++
- b) Warning in both C and C++
- c) Error in C and successful execution in C++
- d) Error in C++ and successful execution in C

View Answer

Answer: d

Explanation: C++ compiler does not allow the programmer to declare a constant variable without initializing it hence the C++ compiler gives an error whereas C allows such declaration, therefore, the program compiles and runs successfully.

3. What happens if the below line is executed in C and C++?

```
int *p = malloc(10);
```

- a) Error in both C and C++
- b) A successful run in both C and C++
- c) Error in C and successful execution in C++
- d) Error in C++ and successful execution in C

View Answer

Answer: d

Explanation: new is a keyword in C++, therefore, we cannot declare a variable with name new but as there is no such keyword new in C, therefore, the program is compiled and executed successfully in C.

3. What happens if the below line is executed in C and C++?

```
int *p = malloc(10);
```

- a) Error in both C and C++
- b) Successful run in both C and C++
- c) Error in C and successful execution in C++
- d) Error in C++ and successful execution in C

View Answer

Answer: d

Explanation: main() function in C++ must return int otherwise the C++ compiler gives the error whereas C does not forces such things on main() function. Thereas when we are making void main(){} function in this program the C++ compiler gives error whereas C compiler runs successfully.

3. What happens if the below line is executed in C and C++?

```
int *p = malloc(10);
```

- a) Error in both C and C++
- b) Outputs Hello twice in both C and C++
- c) Error in C and successful execution in C++
- d) Error in C++ and successful execution in C

View Answer

Answer: a

Explanation: As the func(void) needs no argument during its call, hence when we are calling func(2) with 2 as passed as a parameter then this statement gives the error in both C++ and C compiler.

3. What happens if the below line is executed in C and C++?

```
int *p = malloc(10);
```

- a) Error in both C and C++
- b) Outputs Hello twice in both C and C++
- c) Error in C and Outputs Hello twice in C++
- d) Error in C++ and Outputs Hello twice in C

View Answer

Answer: d

Explanation: In C++ whenever a function without argument is declared it is equivalent to function with void arguments i.e. func() == func(void) whereas in C a function without argument is equivalent to func(...) i.e. it can take any number of arguments so func(2) call is also valid in C but not valid in C++. Hence it gives error in C++ whereas no error in C.

3. What happens if the below line is executed in C and C++?

```
int *p = malloc(10);
```

9. Which of the following type is provided by C++ but not C?

- a) int
- b) bool
- c) float
- d) double

View Answer

Answer: b

Explanation: C++ provides the boolean type to handle true and false values whereas no such type is provided in C.

3. What happens if the below line is executed in C and C++?

```
int *p = malloc(10);
```

10. Which of the following feature is not provided by C?

- a) Pointers
- b) Structures
- c) References
- d) Functions

View Answer

Answer: c

Explanation: References are introduced in C++. They are not present in C.

ayzom.com

C++ Programming Questions and Answers – C++ Concepts – 1

1. Which of the following is not a fundamental type is not present in C but present in C++?

- a) int
- b) float
- c) bool
- d) void

View Answer

Answer: c

Explanation: Boolean type is not present as a fundamental type in C. int type is used as boolean in C whereas in C++ bool is defined as a fundamental type for handling boolean outputs.

2. What is the size of a boolean variable in C++?

- a) 1 bit
- b) 1 byte
- c) 4 bytes
- d) 2 bytes

View Answer

Answer: a

Explanation: Boolean uses only 1 bit as it stores only truth values which can be true(1) or false(0).

3. Which of the following is C++ equivalent for scanf)?

- a) cin
- b) cout
- c) print
- d) input

View Answer

Answer: a

Explanation: C++ uses cin to read input form uses. However C++ also uses scanf).

4. Which of the following is C++ equivalent for printf)?

- a) cin
- b) cout
- c) print
- d) input

View Answer

Answer: b

Explanation: C++ uses cout to print output to console. However C++ also uses printf).

5. Which of the following is the correct difference between cin and scanf)?

- a) both are the same
- b) cin is a stream object whereas scanf) is a function
- c) scanf) is a stream object whereas cin is a function
- d) cin is used for printing whereas scanf) is used for reading input

View Answer

Answer: b

Explanation: cin is a stream object available in C++ whereas scanf) is a function available in both C and C++. both are used for reading input from users.

6. Which of the following is an exit-controlled loop?

- a) for
- b) while
- c) do-while
- d) all of the mentioned

View Answer

Answer: c

Explanation: do-while is called exit controlled loop because in do-while termination condition is checked when we have executed the body of the loop i.e. we are exiting the body and then checking the condition, therefore, it is called exit controlled loop.

7. Which of the following is an entry-controlled loop?

- a) for
- b) while
- c) do-while
- d) both while and for

View Answer

Answer: d

Explanation: Both while and for loops are called entry controlled loop because in both of them the termination condition is checked before we enter the body of the loop hence they are called entry controlled loop.

8. In which part of the for loop termination condition is checked?

for(I;II;III)

{IV}

- a) I
- b) II
- c) III
- d) IV

View Answer

Answer: b

Explanation: In II part the termination condition of the for loop is checked.

9. What is dynamic binding?

- a) The process of linking the actual code with a procedural call during run-time
- b) The process of linking the actual code with a procedural call during compile-time
- c) The process of linking the actual code with a procedural call at any-time
- d) All of the mentioned

View Answer

Answer: a

Explanation: Binding of calls and variables with actual code at run-time is called dynamic binding. For example in the concept of polymorphism types are decided are defined during the execution of code which leads to the different function calls depending upon the types used this is called dynamic binding. As the function call is decided during the run-time therefore dynamic binding happens at run-time.

10. What is static binding?

- a) The process of linking the actual code with a procedural call during run-time
- b) The process of linking the actual code with a procedural call during compile-time
- c) The process of linking the actual code with a procedural call at any-time
- d) All of the mentioned

View Answer

Answer: b

Explanation: Binding of calls and variables with actual code at compile-time is called static binding. For example normally whenever we declare a variable we define its type hence compiler knows what type should be binded to that variable i.e. compiler can decide about that variable this is called static binding.

11. What is name mangling in C++?

- a) The process of adding more information to a function name so that it can be distinguished from other functions by the compiler
- b) The process of making common names for all the function of C++ program for better use
- c) The process of changing the names of variable
- d) The process of declaring variables of different types

View Answer

Answer: a

Explanation: Name mangling is the process of adding some more information to a function name so that it can be distinguished from other functions by the compiler. This is used when a programmer uses the concept of function overloading in his/her program.

-
- a) Output in C is 1 and in C++ is 4
 - b) Output in C is 4 and in C++ is 1
 - c) Output in C is 1 and in C++ is 1
 - d) Output in C is 4 and in C++ is 4

View Answer

Answer: b

Explanation: In C a character is stored as int therefore the size of 'a' is printed as 4 whereas in C++ it is stored as char only therefore in C++ it prints 1.

-
- a) Output in C is 1 and in C++ is 4
 - b) Output in C is 4 and in C++ is 1
 - c) Output in C is 1 and in C++ is 1
 - d) Output in C is 4 and in C++ is 4

View Answer

Answer: c

Explanation: Both in C and C++ the type char has same size which is 1. But a character enclosed inside single quotes has different sizes i.e. in case of char a; the size of a will be 1 in both C and C++ but in case of 'a' size will be 4 in case of C but 1 in case of C++.

14. Which of the following syntax for declaring a variable of struct STRUCT can be used in both C and C++?

- a) struct STRUCT S;
- b) STRUCT S;
- c) Both struct STRUCT S; and STRUCT S;
- d) Both C and C++ have different syntax

View Answer

Answer: a

Explanation: C program requires struct keyword while defining a variable of any structure, therefore, we cannot use the second STRUCT S; definition to declare a variable.

15. What if we define the below structure in C and C++?

- a) Error in C but not in C++
- b) Error in C++ but not in C
- c) No error in both C and C++
- d) Error in both C and C++

View Answer

Answer: a

Explanation: The above definition will give an error in C but not in C++ as C does not allow the programmer to give any default values to any member of structure but C++ does allow.

C++ Programming Questions and Answers – C++ Concepts – 2

3. What is the output of the following C++ code?

```
#include<iostream>
using namespace std;
class A
{
    ~A() {
        cout<<"Destructor called\n";
    }
};
int main()
{
    A a;
    return 0;
}
```

1. Which of the following is the scope resolution operator?

- a) .
- b) *
- c) ::
- d) ~

View Answer

Answer: c

Explanation: :: operator is called scope resolution operator used for accessing a global variable from a function which is having the same name as the variable declared in the function.

3. What is the output of the following C++ code?

```
#include<iostream>
using namespace std;
class A
{
    ~A() {
        cout<<"Destructor called\n";
    }
};
int main()
{
    A a;
    return 0;
}
```

- a) 1
- b) 2
- c) 3
- d) 123

View Answer

Answer: a

Explanation: While printing x we are using :: operator hence the reference is given to global variable hence the global variable x = 1 is printed.

3. What is the output of the following C++ code?

```
#include<iostream>
using namespace std;
class A
{
    ~A() {
        cout<<"Destructor called\n";
    }
}
```

```
};
int main()
{
    A a;
    return 0;
}
```

- a) Destructor called
- b) Nothing will be printed
- c) Error
- d) Segmentation fault

View Answer

Answer: c

Explanation: Whenever a destructor is private then one should not define any normal object as it will be destroyed at the end of the program which will call destructor and as destructor is private the program gives error during compile while in case of pointer object the compiler at compile does not know about the object, therefore, does not give compile error. Hence when the destructor is private then the programmer can declare pointer object but cannot declare a normal object.

3. What is the output of the following C++ code?

```
#include<iostream>
using namespace std;
class A
{
    ~A() {
        cout<<"Destructor called\n";
    }
};
int main()
{
    A a;
    return 0;
}
```

- c) Error
- d) Nothing is printed

View Answer

Answer: c

Explanation: The pointer object is created is not deleted hence the destructor for these objects is not called hence nothing is printed on the screen.

3. What is the output of the following C++ code?

```
#include<iostream>
using namespace std;
class A
{
    ~A() {
        cout<<"Destructor called\n";
    }
};
int main()
{
    A a;
    return 0;
}
```

- a) Garbage value
- b) 0
- c) 99
- d) Error

View Answer

Answer: b

Explanation: In C++ all the uninitialized variables are set to 0 therefore the value of all elements of the array is set to 0.

3. What is the output of the following C++ code?

```
#include<iostream>
using namespace std;
class A
{
    ~A() {
        cout<<"Destructor called\n";
    }
};
int main()
{
    A a;
    return 0;
}
```

- a) cin: garbage value
- b) Error
- c) Segmentation fault
- d) Nothing is printed

View Answer

Answer: a

Explanation: cin is a variable hence overrides the cin object. cin >> cin has no meaning so no error.

3. What is the output of the following C++ code?

```
#include<iostream>
using namespace std;
class A
{
    ~A() {
        cout<<"Destructor called\n";
    }
};
int main()
{
    A a;
    return 0;
}
```

7. Which of the following operator has left to right associativity?

- a) Unary operator
- b) Logical not
- c) Array element access
- d) addressof

View Answer

Answer: c

Explanation: Array element has left to right associativity i.e. expressions are evaluated from left to right in case of array element access.

3. What is the output of the following C++ code?

```
#include<iostream>
using namespace std;
class A
{
    ~A() {
        cout<<"Destructor called\n";
    }
};
int main()
{
    A a;
    return 0;
}
```

```
{  
    A a;  
    return 0;  
}
```

8. Which of the following is accessed by a member function of a class?

- a) The object of that class
- b) All members of a class
- c) The public part of a class
- d) The private part of a class

View Answer

Answer: b

Explanation: A member function of a class can access all the members of its class whether they are private, protected or public.

3. What is the output of the following C++ code?

```
#include<iostream>  
using namespace std;  
class A  
{  
    ~A() {  
        cout<<"Destructor called\n";  
    }  
};  
int main()  
{  
    A a;  
    return 0;  
}
```

9. What is the size of a character literal in C and C++?

- a) 4 and 1
- b) 1 and 4
- c) 1 and 1
- d) 4 and 4

View Answer

Answer: a

Explanation: The size of a character literal is 4 in case of C but it is one in case of C++. You can do `printf("%d", (int)sizeof('a'))`; in both C and C++ to check this.

3. What is the output of the following C++ code?

```
#include<iostream>  
using namespace std;  
class A  
{  
    ~A() {  
        cout<<"Destructor called\n";  
    }  
};  
int main()  
{  
    A a;  
    return 0;  
}
```

10. What is the size of a character type in C and C++?

- a) 4 and 1
- b) 1 and 4
- c) 1 and 1
- d) 4 and 4

View Answer

Answer: c

Explanation: The size of a character literal in both C and C++ is 1. You can do `printf("%d", (int)sizeof(char));` in both C and C++ to check this.

3. What is the output of the following C++ code?

```
#include<iostream>
using namespace std;
class A
{
    ~A() {
        cout<<"Destructor called\n";
    }
};
int main()
{
    A a;
    return 0;
}
```

11. Which of the following is correct?

- a) struct tag is required in both C and C++ while declaring an object of the structure
- b) struct is not required in C but required in C++ while declaring an object of the structure
- c) struct is not required in C++ but required in C while declaring an object of the structure
- d) struct tag is not required in both C and C++ while declaring an object of the structure

View Answer

Answer: c

Explanation: C++ does not require struct keyword while declaring an object of the structure whereas in C we require struct tag for declaring an object.

3. What is the output of the following C++ code?

```
#include<iostream>
using namespace std;
class A
{
    ~A() {
        cout<<"Destructor called\n";
    }
};
int main()
{
    A a;
    return 0;
}
```

12. Which of the following is correct?

- a) struct cannot have member function in C but it can in C++
- b) struct cannot have member function in C++ but it can in C
- c) struct cannot have member function in both C and C++
- d) struct can have member function in both C and C++

View Answer

Answer: a

Explanation: struct can have member function in C++ whereas member functions are not allowed in case of C.

3. What is the output of the following C++ code?

```
#include<iostream>
using namespace std;
class A
{
    ~A() {
        cout<<"Destructor called\n";
    }
}
```

```

    }
};
int main()
{
    A a;
    return 0;
}

```

- a) The program runs fine and both prints output "HELLO THIS IS STRUCTURE"
- b) The program gives an error in case of C but runs perfectly in case of C++
- c) The program gives an error in case of C++ but runs perfectly in case of C
- d) The program gives an error in case of both C and C++

View Answer

Answer: b

Explanation: As C does not allow the structure to have member functions, therefore, it gives an error in case of C but as C++ does allow structures to have member functions, therefore, the C++ does not give an error.

3. What is the output of the following C++ code?

```

#include<iostream>
using namespace std;
class A
{
    ~A() {
        cout<<"Destructor called\n";
    }
};
int main()
{
    A a;
    return 0;
}

```

- a) The program runs fine and both prints output "HELLO THIS IS STRUCTURE"
- b) The program gives an error in case of C but runs perfectly in case of C++
- c) The program gives an error in case of C++ but runs perfectly in case of C
- d) The program gives an error in case of both C and C++

View Answer

Answer: b

Explanation: As C does not allow to initialize any member inside the structure, therefore, the program gives error whereas in case of C++ this is allowed therefore the program does not give any error.

3. What is the output of the following C++ code?

```

#include<iostream>
using namespace std;
class A
{
    ~A() {
        cout<<"Destructor called\n";
    }
};
int main()
{
    A a;
    return 0;
}

```

- a) The program runs fine and both prints output "HELLO THIS IS STRUCTURE"
- b) The program gives an error in case of C but runs perfectly in case of C++
- c) The program gives an error in case of C++ but runs perfectly in case of C
- d) The program gives an error in case of both C and C++

View Answer

Answer: b

Explanation: C does not allow the programmer to declare any static members inside a class whether in C++ it is allowed to declare static variables.

ayzom.com

C++ Programming Questions and Answers – C++ Concepts – 3

1. Which of the following statement is correct?

- a) Structure in C allows Constructor definition
- b) Structure in C++ allows Constructor definition
- c) Both allow Constructor definition
- d) C allows constructor definition while C++ does not

View Answer

Answer: b

Explanation: As C does not allow the programmer to define a function inside a structure and constructor itself is a function, therefore, the constructor definition is not allowed in C whereas such definitions are allowed in C++.

- a) The program runs fine and both prints output “HELLO THIS IS STRUCTURE”
- b) The program gives an error in case of C but runs perfectly in case of C++
- c) The program gives an error in case of C++ but runs perfectly in case of C
- d) The program gives an error in case of both C and C++

View Answer

Answer: b

Explanation: Access specifiers like private, protected and the public are used because the OOPs concept and as C is not an Object Oriented language, therefore, access specifiers are not defined in C and hence C gives error whereas C++ does not.

3. Which of the following is correct about this pointer in C++?

- a) this pointer is passed as a hidden argument in all the functions of a class
- b) this pointer is passed as a hidden argument in all non-static functions of a class
- c) this pointer is passed as a hidden argument in all static functions of a class
- d) all of the mentioned

View Answer

Answer: b

Explanation: As static functions are a type of global function for a class so all the object shares the common instance of that static function whereas all the objects have there own instance for non-static functions and hence they are passed as a hidden argument in all the non-static members but not in static members.

4. Which of the following operator is used with this pointer to access members of a class?

- a) .
- b) !
- c) ->
- d) ~

View Answer

Answer: c

Explanation: this pointer is a type of pointer and as we know pointer object uses the arrow(->) operator to access the members of the class, therefore, this pointer uses -> operator.

5. Why this pointer is used?

- a) To access the members of a class which have the same name as local variables in that scope
- b) To access all the data stored under that class
- c) To access objects of other class
- d) All of the mentioned

View Answer

Answer: a

Explanation: this pointer is used to access the members of a class which have the same name as local variables in that part of the code.

6. How many types of polymorphism are there?

- a) 1
- b) 2
- c) 3
- d) 4

[View Answer](#)

Answer: b

Explanation: There are two types of polymorphism in C++ namely compile-time polymorphism and run-time polymorphism.

7. What is the other name of compile-time polymorphism?

- a) Static polymorphism
- b) Dynamic polymorphism
- c) Executing polymorphism
- d) Non-executing polymorphism

[View Answer](#)

Answer: a

Explanation: Compile-time polymorphism is also known as static polymorphism as it is implemented during the compile-time.

8. What is the other name of run-time polymorphism?

- a) Static polymorphism
- b) Dynamic polymorphism
- c) Executing polymorphism
- d) Non-executing polymorphism

[View Answer](#)

Answer: b

Explanation: Run-time polymorphism is also known as dynamic polymorphism as it is implemented during the run-time of the program.

9. Which of the following is correct about static polymorphism?

- a) In static polymorphism, the conflict between the function call is resolved during the compile time
- b) In static polymorphism, the conflict between the function call is resolved during the run time
- c) In static polymorphism, the conflict between the function call is never resolved during the execution of a program
- d) In static polymorphism, the conflict between the function call is resolved only if it required

[View Answer](#)

Answer: a

Explanation: The conflict between which function to call is resolved during the compile time in static polymorphism i.e. before the execution of the program starts.

10. Which of the following is correct about dynamic polymorphism?

- a) In dynamic polymorphism, the conflict between the function call is resolved during the compile time
- b) In dynamic polymorphism, the conflict between the function call is resolved during the run time
- c) In dynamic polymorphism, the conflict between the function call is never resolved during the execution of program
- d) In dynamic polymorphism, the conflict between the function call is resolved at the beginning of the program

[View Answer](#)

Answer: b

Explanation: The conflict between which function to call is resolved during the run time in dynamic polymorphism i.e. the conflict is resolved when the execution reaches the function call statement.

11. Which of the following operator(s) can be used with pointers?

- i) – only
- ii) +, *
- iii) +, –
- iv) +, –, *

v) /
vi) + only
a) i only
b) vi only
c) ii and v
d) iv
View Answer

Answer: a

Explanation: The only arithmetic operator that can be used with a pointer is – subtraction operator. No arithmetic operator can be used with pointers.

12. What is std in C++?

a) std is a standard class in C++
b) std is a standard namespace in C++
c) std is a standard header file in C++
d) std is a standard file reading header in C++
View Answer

Answer: b

Explanation: std is a standard namespace present in C++ which contains different stream classes and objects like cin, cout, etc. and other standard functions.

a) Hellow World
b) Compile-time error
c) Run-time error
d) Segmentation fault
View Answer

Answer: b

Explanation: cout is defined under the namespace std and without including std namespace we cannot cout, therefore, the program gives an error.

14. Which of the following syntax can be used to use a member of a namespace without including that namespace?

a) namespace::member
b) namespace->member
c) namespace.member
d) namespace~member
View Answer

Answer: a

Explanation: To use a member of a namespace without including the namespace is done by this syntax namespace::member.

a) Both code 1 and code 2
b) Code 1 only
c) Code 2 only
d) Neither code 1 nor code 2
View Answer

Answer: d

Explanation: Neither code 1 nor code2 will give error as both are syntactically correct as in first code we have included namespace std and in second one we have used scope resolution operator to resolve the conflict.

C++ Programming Questions and Answers – Static Constant Keyword

a) 0 0

b) 5 0

c) 0 5

d) 5 5

View Answer

Answer: c

Explanation: Static function can be called without using objects therefore the first call is fine. Next when we are creating 5 objects of the class then value of x is updated each time and as static variables are global to class therefore each updation of x is reflected back to each object hence value of x is 5.

a) 1 2 3

b) 2 2 2

c) 1 3 1

d) 1 1 1

View Answer

Answer: a

Explanation: In this as next_id is a static variable so and initialized with 1 therefore the id value for 1st objects is 1 and next_id is updated to 2. In this way next_id is assigned to id in each object creation and updated by 1 so in this way value of each Id is updated.

3. Which of the following is correct about static variables?

a) Static functions do not support polymorphism

b) Static data members cannot be accessed by non-static member functions

c) Static data members functions can access only static data members

d) Static data members functions can access both static and non-static data members

View Answer

Answer: c

Explanation: Static member functions can access static data members only. Static member functions can be overloaded. Static data members can be accessed by non-static member functions.

a) Garbage value

b) Compile-time Error

c) Run-time Error

d) Nothing is printed

View Answer

Answer: b

Explanation: Every static member function of a class must be initialized explicitly before use and a data member, a of class A declared inside class B is used without initializing 'a' therefore the program gives an error.

a) 4 4 4 4

b) 1 2 3 4

c) 1 1 1 1

d) 0 1 2 3

View Answer

Answer: b

Explanation: Here we are returning the reference of object by the function call fun() therefore this type of call is allowed. Also as count is static member of the class therefore updation is reflected to the whole class and to every object of the class. Therefore the four function calls to fun() function updates the value of count and prints.

- a) fun() is static
- b) Compile-time Error
- c) Run-time Error
- d) Nothing is printed

View Answer

Answer: b

Explanation: The prototype of the functions are not matched. The function declared inside a class does not have static linkage however the class defined outside the class has the static linkage, therefore, the program gives an error.

7. Const qualifier can be applied to which of the following?

- i) Functions inside a class
- ii) Arguments of a function
- iii) Static data members
- iv) Reference variables

- a) i, ii and iii
- b) i, ii, iii, and iv
- c) ii, iii and iv
- d) i only

View Answer

Answer: b

Explanation: const keyword can be applied to all of the following mentioned above.

- a) 0 0
- b) Garbage values
- c) Compile error
- d) Segmentation fault

View Answer

Answer: c

Explanation: C++ does not allow a constant object to access any non constant member functions and as getY() is a non constant function and t is a constant object therefore the program gives the error.

- a) 10
- b) Garbage value
- c) Error
- d) Segmentation fault

View Answer

Answer: c

Explanation: In C++, all the constant variables must be initialized while declaration and they cannot be modified later in the program. Now in this program as we have declared the constant variable x in first line and initializing it in the next line therefore the program gives the error.

- a) 9
- b) Garbage value
- c) Error
- d) Segmentation fault

View Answer

Answer: a

Explanation: The program is syntactically and semantically correct hence the program is compiled and executed successfully.

C++ Programming Questions and Answers – Types

1. What is the size of `wchar_t` in C++?

- a) 2
- b) 4
- c) 2 or 4
- d) Based on the number of bits in the system

[View Answer](#)

Answer: d

Explanation: Compiler wants to make CPU as more efficient in accessing the next value.

2. Pick the odd one out

- a) array type
- b) character type
- c) boolean type
- d) integer type

[View Answer](#)

Answer: a

Explanation: Array type is not the basic type and it is constructed using the basic type.

3. Which data type is used to represent the absence of parameters?

- a) int
- b) short
- c) void
- d) float

[View Answer](#)

Answer: c

Explanation: Because void specifies an empty set of values/parameters.

4. What does `'\a'` escape code represent?

- a) alert
- b) backslash
- c) tab
- d) form feed

[View Answer](#)

Answer: a

Explanation: Because `\a` is used to produce a beep sound.

5. Which type is best suited to represent the logical values?

- a) integer
- b) boolean
- c) character
- d) all of the mentioned

[View Answer](#)

Answer: b

Explanation: Logical values can be either true or false, so the boolean type is suited for it.

6. Identify the user-defined types from the following?

- a) enumeration
- b) classes

- c) both enumeration and classes
- d) int

View Answer

Answer: c

Explanation: They must be defined by the users before use, unlike the other types which are readily available.

- a) f is a function taking an argument of type int and returning a floating point number
- b) f is a function taking an argument of type float and returning an integer
- c) f is a function of type float
- d) none of the mentioned

View Answer

Answer: b

Explanation: The argument that is passed to a function f is of float type and the function finally returns a value that is of integer type.

8. The value 132.54 can be represented using which data type?

- a) double
- b) void
- c) int
- d) bool

View Answer

Answer: a

Explanation: The given value is with decimal points, so float or double can be used.

9. When a language has the capability to produce new data type mean, it can be called as

- a) overloaded
- b) extensible
- c) encapsulated
- d) reprehensible

View Answer

Answer: b

Explanation: Extensible is used to add new features to C++.

10. Pick the odd one out.

- a) integer, character, boolean, floating
- b) enumeration, classes
- c) integer, enum, void
- d) arrays, pointer, classes

View Answer

Answer: c

Explanation: Option a consists of all fundamental types, option b consists of user-defined types and option d consists of derived types but option c is a mixture.

C++ Programming Questions and Answers – Booleans

3. What is the value of the bool?

```
bool is_int(789.54)
```

1. Is bool a fundamental data type in C++?

- a) Yes
- b) No, it is a typedef of unsigned char
- c) No, it is an enum of {false, true}
- d) No, it is expanded from macros

View Answer

Answer: a

Explanation: C++ has bool as a fundamental data type.

3. What is the value of the bool?

```
bool is_int(789.54)
```

2. Find the odd one out:

- a) `std::vector<int>`
- b) `std::vector<short>`
- c) `std::vector<long>`
- d) `std::vector<bool>`

View Answer

Answer: d

Explanation: `std::vector<bool>` is a specialized version of vector, which is used for elements of type bool and optimizes for space. It behaves like the unspecialized version of vector and the storage is not necessarily an array of bool values, but the library implementation may optimize storage so that each value is stored in a single bit.

3. What is the value of the bool?

```
bool is_int(789.54)
```

- a) True
- b) False
- c) 1
- d) None of the mentioned

View Answer

Answer: b

Explanation: The given number is a double not an integer, so the function returns 0 which is boolean false.

3. What is the value of the bool?

```
bool is_int(789.54)
```

4. What happens when a null pointer is converted into bool?

- a) an error is flagged
- b) bool value evaluates to true
- c) bool value evaluates to false
- d) the statement is ignored

View Answer

Answer: c

Explanation: A pointer can be implicitly converted to a bool. A nonzero pointer converts to true and zero valued pointer converts to false.

3. What is the value of the bool?

```
bool is_int(789.54)
```

5. Which of the following statements are false?

- a) bool can have two values and can be used to express logical expressions
- b) bool cannot be used as the type of the result of the function
- c) bool can be converted into integers implicitly
- d) a bool value can be used in arithmetic expressions

View Answer

Answer: b

Explanation: Boolean can be used as a return value of a function.

3. What is the value of the bool?

```
bool is_int(789.54)
```

6. For what values of the expression is an if-statement block not executed?

- a) 0 and all negative values
- b) 0 and -1
- c) 0
- d) 0, all negative values, all positive values except 1

View Answer

Answer: c

Explanation: The if-statement block is only not executed when the expression evaluates to 0. its just syntactic sugar for a branch-if-zero instruction.

3. What is the value of the bool?

```
bool is_int(789.54)
```

7. Which of the two operators ++ and — work for the bool data type in C++?

- a) None
- b) ++
- c) —
- d) ++ & —

View Answer

Answer: b

Explanation: Due to the history of using integer values as booleans, if an integer is used as a boolean, then incrementing will mean that whatever its truth value before the operation, it will have a truth-value of true after it. However, it is not possible to predict the result of — given knowledge only of the truth value of x, as it could result in false.

3. What is the value of the bool?

```
bool is_int(789.54)
```

- a) 55
- b) 62
- c) 52
- d) none of the mentioned

View Answer

Answer: c

Explanation: In this question, value of x = true and value of y will be also true as f(a,b) will return a non-zero value. Now when adding these values with integers, the implicit type conversion takes place hence converting both x and y to 1(integer equivalent of bool true value). So

expression $(a*b) + (x+y)$ is evaluated to 52.

3. What is the value of the bool?

```
bool is_int(789.54)
```

- a) 0
- b) 16
- c) 12
- d) 2

View Answer

Answer: b

Explanation: | means bitwise OR operation so $x | y$ (0101 | 1010) will be evaluated to 1111 which is integer 15 and as a is true and b is false so $a+b(1+0) = 1$. So final value of expression in line #10 will be $15 + 1 = 16$.

3. What is the value of the bool?

```
bool is_int(789.54)
```

10. Evaluate the following

$(\text{false} \ \&\& \ \text{true}) \ || \ \text{false} \ || \ \text{true}$

- a) 0
- b) 1
- c) false
- d) none of the mentioned

View Answer

Answer: b

Explanation: The given expression is equivalent to

$[(\text{false AND True}) \text{ OR false OR true}]$ This is OR or three values so if any of them will be true then whole exp will be true and as we have last value as true so the answer of expression will be TRUE.

C++ Programming Questions and Answers – Character Types

1. How many characters are specified in the ASCII scheme?

- a) 64
- b) 128
- c) 256
- d) none of the mentioned

View Answer

Answer: b

Explanation: There are 128 characters defined in the C++ ASCII list.

2. Select the right option.

Given the variables p, q are of char type and r, s, t are of int type

- 1. $t = (r * s) / (r + s);$
- 2. $t = (p * q) / (r + s);$
- a) 1 is true but 2 is false
- b) 1 is false and 2 is true
- c) both 1 and 2 are true
- d) both 1 and 2 are false

View Answer

Answer: c

Explanation: Every character constant has an integer value. Also char belongs to the integral type hence arithmetic and logical operations can be performed on them.

3. Which of the following belongs to the set of character types?

- a) char
- b) wchar_t
- c) only a
- d) both wchar_t and char

View Answer

Answer: d

Explanation: wchar_t and char is used to represent wide character and character.

- a) A
- b) N
- c) J
- d) I

View Answer

Answer: c

Explanation: The literal value for 74 is J. So it will be printing J.

5. How do we represent a wide character of the form wchar_t?

- a) L'a'
- b) l'a'
- c) L[a].
- d) la

View Answer

Answer: a

Explanation: A wide character is always indicated by immediately preceding the character literal by an L.

- a) Compiler error
- b) 12
- c) 10
- d) Empty

View Answer

Answer: c

Explanation: The value '\012' means the character with value 12 in octal, which is decimal 10.

7. In C++, what is the sign of character data type by default?

- a) Signed
- b) Unsigned
- c) Implementation dependent
- d) None of the mentioned

View Answer

Answer: c

Explanation: The standard does not specify if plain char is signed or unsigned. There are three distinct character types according to the standard: char, signed char and unsigned char.

8. Is the size of character literals different in C and C++?

- a) Implementation defined
- b) Can't say
- c) Yes, they are different
- d) No, they are not different

View Answer

Answer: c

Explanation: In C++, `sizeof('a') == sizeof(char) == 1`. In C however, `sizeof('a') == sizeof(int)`.

9. Suppose in a hypothetical machine, the size of char is 32 bits. What would `sizeof(char)` return?

- a) 4
- b) 1
- c) Implementation dependent
- d) Machine dependent

View Answer

Answer: b

Explanation: The standard does NOT require a char to be 8-bits, but does require that `sizeof(char)` return 1.

10. What constant defined in `<climits>` header returns the number of bits in a char?

- a) `CHAR_SIZE`
- b) `SIZE_CHAR`
- c) `BIT_CHAR`
- d) `CHAR_BIT`

View Answer

Answer: d

Explanation: `CHAR_BIT` is a macro constant defined in `<climits>` header file which expresses number of bits in a character object in bytes.

C++ Programming Questions and Answers – Integer Types

1. The `size_t` integer type in C++ is?

- a) Unsigned integer of at least 64 bits
- b) Signed integer of at least 16 bits
- c) Unsigned integer of at least 16 bits
- d) Signed integer of at least 64 bits

View Answer

Answer: c

Explanation: The `size_t` type is used to represent the size of an object. Hence, it's always unsigned. According to the language specification, it is at least 16 bits.

- a) x is greater
- b) y is greater
- c) implementation defined
- d) arbitrary

View Answer

Answer: a

Explanation: x is promoted to unsigned int on comparison. On conversion x has all bits set, making it the bigger one.

3. Which of these expressions will return true if the input integer v is a power of two?

- a) $(v \mid (v + 1)) == 0;$
- b) $(\sim v \ \& \ (v - 1)) == 0;$
- c) $(v \mid (v - 1)) == 0;$
- d) $(v \ \& \ (v - 1)) == 0;$

View Answer

Answer: d

Explanation: Power of two integers have a single set bit followed by unset bits.

- a) 1
- b) -1
- c) 127
- d) Implementation defined

View Answer

Answer: d

Explanation: Right shift of signed integers is undefined, and has implementation-defined behaviour.

5. Which of these expressions will make the rightmost set bit zero in an input integer x?

- a) $x = x \mid (x-1)$
- b) $x = x \ \& \ (x-1)$
- c) $x = x \mid (x+1)$
- d) $x = x \ \& \ (x+2)$

View Answer

Answer: b

Explanation: If x is odd the last bit will be 1 and last bit of x-1 will become 0. If x is even then last bit of x will be 0 and last bit of x-1 will become 1. In both case AND operation of 1 and 0 will be 0. Hence last bit of final x will be 0.

6. Which of these expressions will isolate the rightmost set bit?

- a) $x = x \ \& \ (\sim x)$
- b) $x = x \wedge (\sim x)$

c) $x = x \& (-x)$

d) $x = x \wedge (-x)$

View Answer

Answer: c

Explanation: Negative of a number is stored as 2's complement in C++, so when you will take AND of x and $(-x)$ the rightmost digit will be preserved.

7. 0946, 786427373824, 'x' and 0X2f are _____ and _____ literals respectively.

a) decimal, character, octal, hexadecimal

b) octal, hexadecimal, character, decimal

c) hexadecimal, octal, decimal, character

d) octal, decimal, character, hexadecimal

View Answer

Answer: d

Explanation: Literal integer constants that begin with 0x or 0X are interpreted as hexadecimal and the ones that begin with 0 as octal. The character literal are written within "".

a) ANDing integer 'a' with 'true' :8

b) ANDing integer 'a' with 'true' :0

c) ANDing integer 'a' with 'true' :1

d) None of the mentioned

View Answer

Answer: a

Explanation: The && operator in C++ uses short-circuit evaluation so that if bool1 evaluates to false it doesn't bother evaluating bool2. So as here bool1 is 8 which is true as it is non-zero so C++ does not care about the expression further and prints the answer of expression which is 8. If you write true && 8 then the output will be 1 because true is true and its integer equivalent is 1 so 1 will be printed.

a) compile time error

b) -1 1

c) 1 -1

d) implementation defined

View Answer

Answer: b

Explanation: Sign of result of mod operation on negative numbers is sign of the dividend.

a) 0

b) 1

c) Compiler error may be possible

d) Runtime error may be possible

View Answer

Answer: c

Explanation: Using & on a register variable may be invalid, since the compiler may store the variable in a register, and finding the address of it is illegal.

C++ Programming Questions and Answers – Floating Point Types

1. Which of the following is not one of the sizes of the floating point types?

- a) short float
- b) float
- c) long double
- d) double

View Answer

Answer: a

Explanation: Floating point types occur in only three sizes-float, long double and double.

2. Which of the following is a valid floating-point literal?

- a) f287.333
- b) F287.333
- c) 287.e2
- d) 287.3.e2

View Answer

Answer: c

Explanation: To make a floating point literal, we should attach a suffix of 'f' or 'F' and there should not be any blank space.

3. What is the range of the floating point numbers?

- a) -3.4E+38 to +3.4E+38
- b) -3.4E+38 to +3.4E+34
- c) -3.4E+38 to +3.4E+36
- d) -3.4E+38 to +3.4E+32

View Answer

Answer: a

Explanation: This is the defined range of floating type number in C++. Also range for +ve and -ve side should be same so the answer is -3.4E+38 to +3.4E+38.

4. Which of three sizes of floating point types should be used when extended precision is required?

- a) float
- b) double
- c) long double
- d) extended float

View Answer

Answer: c

Explanation: Float for single precision, double for double precision and long double for extended precision.

- a) harvard
- b) stanford
- c) compile time error
- d) runtime error

View Answer

Answer: a

Explanation: Float stores floating point numbers with 8 place accuracy and requires 4 bytes of Memory. Double has 16 place accuracy having the size of 8 bytes.

Output:

```
$ g++ float3.cpp
$ a.out
```

harvard

- a) 0.11
 - b) 0.10000000000000001
 - c) 0.100001
 - d) compile time error
- View Answer

Answer: b

Explanation: The double had to truncate the approximation's due to its limited memory, which resulted in a number that is not exactly 0.1.

Output:

```
$ g++ float2.out
$ a.out
0.10000000000000001
```

- a) 123.00
 - b) 1.23
 - c) 123
 - d) compile time error
- View Answer

Answer: c

Explanation: The value 123 is printed because of its precision.

```
$ g++ float.cpp
$ a.out
123
```

8. Which is used to indicate single precision value?

- a) F or f
- b) L or l
- c) Either F or for L or l
- d) Neither F or for L or l

View Answer

Answer: a

Explanation: Either F or f can be used to indicate single precision values.

- a) equal
- b) not equal
- c) compile time error
- d) runtime error

View Answer

Answer: a

Explanation: 0.5f results in 0.5 to be stored in floating point representations.

Output:

```
$ g++ float.cpp
$ a.out
equal
```

10. Which is correct with respect to size of the data types?

- a) char > int < float
- b) int < char > float
- c) char < int < float
- d) char < int < double

View Answer

Answer: d

Explanation: The char has less bytes than int and int has less bytes than double whereas int and float can potentially have same sizes.

ayzom.com

C++ Programming Questions and Answers – Sizes

1. The size of an object or a type can be determined using which operator?

- a) malloc
- b) sizeof
- c) malloc
- d) calloc

View Answer

Answer: b

Explanation: The sizeof operator gives the size of the object or type.

2. It is guaranteed that a ____ has at least 8 bits and a ____ has at least 16 bits.

- a) int, float
- b) char, int
- c) bool, char
- d) char, short

View Answer

Answer: d

Explanation: char types in C++ require atleast 8 bits and short requires atleast 16 bits, whereas for bool only 1 bit suffices and both int and float requires atleast 32 bits.

3. Implementation dependent aspects about an implementation can be found in ____

- a) <implementation>
- b) <limits>
- c) <limit>
- d) <numeric>

View Answer

Answer: b

Explanation: The limit header holds the details of the machine dependent details.

4. Size of C++ objects are expressed in terms of multiples of the size of a ____ and the size of a char is ____

- a) char, 1
- b) int, 1
- c) float, 8
- d) char, 4

View Answer

Answer: a

Explanation: Each object in C++ is expressed in terms of char type and size of char type is one byte.

5. Identify the incorrect option.

- a) $1 \leq \text{sizeof}(\text{bool}) \leq \text{sizeof}(\text{long})$
- b) $\text{sizeof}(\text{float}) \leq \text{sizeof}(\text{double}) \leq \text{sizeof}(\text{long double})$
- c) $\text{sizeof}(\text{char}) \leq \text{sizeof}(\text{long}) \leq \text{sizeof}(\text{wchar_t})$
- d) $\text{sizeof}(N) = \text{sizeof}(\text{signed } N) = \text{sizeof}(\text{unsigned } N)$

View Answer

Answer: c

Explanation: $\text{sizeof}(\text{char}) \leq \text{sizeof}(\text{wchar_t}) \leq \text{sizeof}(\text{long})$.

- a) 2
- b) 4

c) Depends on compiler

d) Garbage

View Answer

Answer: c

Explanation: The sum of three numbers are belongs to different number systems, so the result is type casted into integer.

Output:

```
$ g++ size.cpp
$ a.out
4
```

a) 4

b) 2

c) 8

d) garbage

View Answer

Answer: c

Explanation: The size of the double data type is 8.

```
$ g++ size1.cpp
$ a.out
8
```

a) 2

b) 4

c) 8

d) garbage

View Answer

Answer: b

Explanation: In this program, integer is converted into float. Therefore the result of num1 and num2 is float. And it is returning the size of the float.

Output:

```
$ g++ size2.cpp
$ a.out
4
```

a) 2 6

b) 4 6

c) 2 5

d) 4 5

View Answer

Answer: d

Explanation: The a as a integer will be converted to float while calculating the size. The value of any variable doesn't modify inside sizeof operator. Hence value of variable a will remain 5.

Output:

```
$ g++ size3.cpp
$ a.out
4 5
```

a) 1 4 4

b) 1 4 8

c) 1 8 8

d) none of the mentioned

View Answer

Answer: a

Explanation: Character is 1 byte, integer 4 bytes and float 4 bytes.

ayzom.com

C++ Programming Questions and Answers – Void

3. What does the following statement mean?

```
void a;
```

1. Which of the following will not return a value?

- a) null
- b) void
- c) empty
- d) free

View Answer

Answer: b

Explanation: Because void represents empty set of values so nothing will be return.

3. What does the following statement mean?

```
void a;
```

2. ____ have the return type void?

- a) all functions
- b) constructors
- c) destructors
- d) none of the mentioned

View Answer

Answer: d

Explanation: Constructor creates an Object and Destructor destroys the object. They are not supposed to return anything, not even void.

3. What does the following statement mean?

```
void a;
```

- a) variable a is of type void
- b) a is an object of type void
- c) declares a variable with value a
- d) flags an error

View Answer

Answer: d

Explanation: There are no void objects.

3. What does the following statement mean?

```
void a;
```

4. Choose the incorrect option

- a) void is used when the function does not return a value
- b) void is also used when the value of a pointer is null
- c) void is used as the base type for pointers to objects of unknown type
- d) void is a special fundamental type

View Answer

Answer: b

Explanation: void fundamental type is used in the cases of a and c.

3. What does the following statement mean?

`void a;`

- a) 20
- b) compile time error
- c) runtime error
- d) none of the mentioned

View Answer

Answer: b

Explanation: void will not accept any values to its type.

ayzom.com

C++ Programming Questions and Answers – Enumerations

1. Identify the incorrect option.

- a) enumerators are constants
- b) enumerators are user-defined types
- c) enumerators are same as macros
- d) enumerator values start from 0 by default

View Answer

Answer: c

Explanation: Enumerators are used in order to create our own types whereas macros are textual substitutions.

2. In which type do the enumerators are stored by the compiler?

- a) string
- b) integer
- c) float
- d) none of the mentioned

View Answer

Answer: b

Explanation: In C++, enumerations are stored as integers by the compiler starting with 0.

3. To which of these enumerators can be assigned?

- a) integer
- b) negative
- c) enumerator
- d) all of the mentioned

View Answer

Answer: d

Explanation: Since enumerators evaluate to integers, and integers can be assigned to enumerators, enumerators can be assigned to other enumerators.

4. What will happen when defining the enumerated type?

- a) it will not allocate memory
- b) it will allocate memory
- c) it will not allocate memory to its variables
- d) none of the mentioned

View Answer

Answer: a

Explanation: Enumerator will allocate the memory when its variables are defined.

5. Which variable does equals in size with enum variable?

- a) int variable
- b) float variable
- c) string variable
- d) none of the mentioned

View Answer

Answer: a

Explanation: The enum variable is converted to an integer and stored by the compiler. So both are equal in size.

- a) If you were cat, you would be 5
- b) If you were cat, you would be 2

c) If you were cat, you would be 7

d) None of the mentioned

View Answer

Answer: b

Explanation: The age will be divided by using compound assignment operator and so it will return the age of the cat according to your age.

```
$ g++ enum1.cpp
$ a.out
If you were cat, you would be 2
```

a) 323334

b) 323232

c) 323130

d) none of the mentioned

View Answer

Answer: a

Explanation: If we not assigned any value to enum variable means, then the next number to initialized number will be allocated to the variable.

Output:

advertisement

```
$ g++ enum2.cpp
$ a.out
323334
```

a) 012345

b) 123456

c) compile time error

d) runtime error

View Answer

Answer: a

Explanation: The enumerator values start from zero if it is unassigned.

Output:

```
$ g++ enum3.cpp
$ a.out
012345 <pre>[/expand]
```

9. What is the output of this program?

```
<pre lang="cpp" line="1" cssfile="hkl_style">
#include <iostream>
using namespace std;
int main()
{
    enum channel {star, sony, zee};
    enum symbol {hash, star};
    int i = 0;
    for (i = star; i <= zee; i++) {
        printf("%d ", i);
    }
    return 0;
}
```

a) 012

b) 123

c) compile time error

d) runtime error

View Answer Answer: c

Explanation: Enumeration variable 'star' appears two times in main() which causes the error. An enumeration constant must be unique within the

scope.

- a) 01234567891011
- b) 123456789101112
- c) 34567891011
- d) 123456789

View Answer

Answer: c

Explanation: We are getting the values from march to november and printing its concern number.

ayzom.com

C++ Programming Questions and Answers – Declaration

- a) both 1 and 2 declare i
- b) 1 declares the variable i and 2 defines i
- c) 1 declares and defines i, 2 declares i
- d) 1 declares i, 2 declares and defines i

View Answer

Answer: d

Explanation: The keyword extern is not a definition and is not allocated storage until it is initialized.

- a) Statement 1 is true, Statement 2 is false
- b) Statement 2 is true, Statement 1 is false
- c) Both are false
- d) Both are true

View Answer

Answer: b

Explanation: An identifier can be declared many times must be defined just once.

3. Which of the given statements are false.

- 1. extern int func;
- 2. extern int func2(int,int);
- 3. int func2(int,int);
- 4. extern class foo;

- a) 3 and 4 only
- b) 2 and 3 only
- c) only 4
- d) 2, 3 and 4

View Answer

Answer: c

Explanation: No extern are allowed for class declarations.

- a) Statement 1 is true, Statement 2 is false
- b) Statement 2 is true, Statement 1 is false
- c) Both are false
- d) Both are true

View Answer

Answer: c

Explanation: Global values are implicitly initialised to 0, but local values have to be initialised by the system.

- a) 2035655065
- b) 2035655035
- c) 2035635065
- d) none of the mentioned

View Answer

Answer: a

Explanation: The local values of a and g within the block are more dominant than the global values.

Output:

```
$ g++ decl.cpp
```

```
$ a.out
```

```
2035655065
```

6. Can two functions declare variables(non static) with the same name.

- a) No
- b) Yes
- c) Yes, but not a very efficient way to write programs
- d) No, it gives a runtime error

View Answer

Answer: c

Explanation: We can declare variables with the same name in two functions because their scope lies within the function.

- a) 234
- b) 111
- c) 123
- d) 235

View Answer

Answer: a

Explanation: The variable that is declared as static has a file scope.

Output:

advertisement

```
$ g++ dec2.cpp
$ a.out
234
```

- a) 0123456789
- b) 123456789
- c) 0
- d) error

View Answer

Answer: d

Explanation: We will get compilation error because 'i' is an undeclared identifier.

- a) Statement 1 and 2 are wrong
- b) Statement 2 and 3 are wrong
- c) Statement 1 and 3 are wrong
- d) All the three are wrong

View Answer

Answer: c

Explanation: In statement 1 lvalue is required as unary '&' operand and in statement 3 lvalue is required as left operand.

- a) char*
- b) char
- c) CHAR
- d) unknown

View Answer

Answer: a

Explanation: The statement makes CHAR a synonym for char*.

C++ Programming Questions and Answers – Pointers

- a) pointer to a pointer
 - b) pointer to an array of chars
 - c) pointer to function taking a char* argument and returns an int
 - d) function taking a char* argument and returning a pointer to int
- View Answer

Answer: c

Explanation: The (*fn) represents a pointer to a function and char* as arguments and returning int from the function. So according to that, the above syntax represents a pointer to a function taking a char* as an argument and returning int.

2. The operator used for dereferencing or indirection is ____
- a) *
 - b) &
 - c) ->
 - d) -->>
- View Answer

Answer: a

Explanation: * is used as dereferencing operator, used to read value stored at the pointed address.

3. Choose the right option
- string* x, y;
- a) x is a pointer to a string, y is a string
 - b) y is a pointer to a string, x is a string
 - c) both x and y are pointers to string types
 - d) none of the mentioned
- View Answer

Answer: a

Explanation: * is to be grouped with the variables, not the data types.

4. Which one of the following is not a possible state for a pointer.
- a) hold the address of the specific object
 - b) point one past the end of an object
 - c) zero
 - d) point to a type
- View Answer

Answer: d

Explanation: A pointer can be in only 3 states a,b and c.

5. Which of the following is illegal?
- a) int *ip;
 - b) string s, *sp = 0;
 - c) int i; double* dp = &i;
 - d) int *pi = 0;
- View Answer

Answer: c

Explanation: dp is initialized int value of i.

- a) b is assigned to a
- b) p now points to b

- c) a is assigned to b
- d) q now points to a

View Answer

Answer: b

Explanation: Assigning to reference changes the object to which the reference is bound.

- a) 5
- b) 10
- c) 15
- d) it will return some random number

View Answer

Answer: d

Explanation: Array element cannot be address of auto variable. It can be address of static or extern variables.

8. The correct statement for a function that takes pointer to a float, a pointer to a pointer to a char and returns a pointer to a pointer to a integer is

- a) `int **fun(float**, char**)`
- b) `int *fun(float*, char*)`
- c) `int **fun(float*, char**)`
- d) `int ***fun(*float, **char)`

View Answer

Answer: c

Explanation: None.

- a) ABCDEFGHIJ
- b) AAAAAAAAAA
- c) JJJJJJJJ
- d) None of the mentioned

View Answer

Answer: a

Explanation: Each time we are assigning $65 + i$. In first iteration $i = 0$ and 65 is assigned. So it will print from A to J.

```
$ g++ point1.cpp
```

```
$ a.out
```

```
ABCDEFGHIJ
```

- a) fg
- b) cdef
- c) defg
- d) abcd

View Answer

Answer: a

Explanation: Pointer ptr points to string 'fg'. So it prints fg.

Output:

```
$ g++ point.cpp
```

```
$ a.out
```

```
fg
```

C++ Programming Questions and Answers – Arrays

1. Which of the following correctly declares an array?

- a) `int array[10];`
- b) `int array;`
- c) `array{10};`
- d) `array array[10];`

View Answer

Answer: a

Explanation: Because array variable and values need to be declared after the datatype only.

2. What is the index number of the last element of an array with 9 elements?

- a) 9
- b) 8
- c) 0
- d) Programmer-defined

View Answer

Answer: b

Explanation: Because the first element always starts at 0. So it is on 8 position.

3. What is the correct definition of an array?

- a) An array is a series of elements of the same type in contiguous memory locations
- b) An array is a series of element
- c) An array is a series of elements of the same type placed in non-contiguous memory locations
- d) None of the mentioned

View Answer

Answer: a

Explanation: Correct definition of an array is An array is a series of elements of the same type in contiguous memory locations.

4. Which of the following accesses the seventh element stored in array?

- a) `array[6];`
- b) `array[7];`
- c) `array(7);`
- d) `array;`

View Answer

Answer: a

Explanation: The array location starts from zero, So it can accessed by `array[6]`.

5. Which of the following gives the memory address of the first element in array?

- a) `array[0];`
- b) `array[1];`
- c) `array(2);`
- d) `array;`

View Answer

Answer: d

Explanation: To get the address of ith index of an array, we use following syntax `(arr + i)`. So as we need address of first index we will use `(arr + 0)` equivalent to `arr`.

a) 6553

b) 6533

c) 6522
d) 12200
View Answer

Answer: b

Explanation: In this program we are adding the every element of two arrays. Finally we got output as 6533.

Output:

```
$ g++ array.cpp
$ a.out
6533
```

a) 25
b) 26
c) 27
d) None of the mentioned
View Answer

Answer: d

Explanation: We are adding all the elements in the array and printing it. Total elements in the array is 7, but our for loop will go beyond 7 and add a garbage value.

a) 15
b) 18
c) garbage value
d) compile time error
View Answer

Answer: d

Explanation: The conversion is invalid in this array. So it will arise error. The following compilation error will be raised:
cannot convert from 'int *' to 'int'

This is because &a,&b and &c represent int* whereas the array defined is of int type.

a) ABC
b) ABCD
c) AB
d) None of the mentioned
View Answer

Answer: a

Explanation: We are just printing the values of first 3 values.

```
$ g++ array.cpp
$ a.out
ABC
```

a) -15
b) -30
c) compile time error
d) garbage value
View Answer

Answer: b

Explanation: It's just printing the negative value of the concern element.

```
$ g++ array.cpp
$ a.out
-30
```

C++ Programming Questions and Answers – Pointers into Arrays

3. What is the output of this program?

```
1.  #include <iostream>
2.  using namespace std;
3.  int main()
4.  {
5.      int a[2][4] = {3, 6, 9, 12, 15, 18, 21, 24};
6.      cout << *(a[1] + 2) << (*(a + 1) + 2) << 2[1[a]];
7.      return 0;
8.  }
```

1. What is the meaning of the following declaration?

`int(*p[5])();`

- a) p is pointer to function
- b) p is array of pointer to function
- c) p is pointer to such function which return type is the array
- d) p is pointer to array of function

View Answer

Answer: b

Explanation: In the above declaration the variable p is the array, not the pointer.

3. What is the output of this program?

```
1.  #include <iostream>
2.  using namespace std;
3.  int main()
4.  {
5.      int a[2][4] = {3, 6, 9, 12, 15, 18, 21, 24};
6.      cout << *(a[1] + 2) << (*(a + 1) + 2) << 2[1[a]];
7.      return 0;
8.  }
```

2. What is size of generic pointer in C++ (in 32-bit platform) ?

- a) 2
- b) 4
- c) 8
- d) 0

View Answer

Answer: b

Explanation: Size of any type of pointer is 4 bytes in 32-bit platforms.

3. What is the output of this program?


```

1.  #include <iostream>
2.  using namespace std;
3.  int main()
4.  {
5.      int a[2][4] = {3, 6, 9, 12, 15, 18, 21, 24};
6.      cout << *(a[1] + 2) << *((a + 1) + 2) << 2[1[a]];
7.      return 0;
8.  }

```

- a) 15 18 21
- b) 21 21 21
- c) 24 24 24
- d) Compile time error

View Answer

Answer: b

Explanation: $a[1][2]$ means $1 * (4) + 2 = 6$ th element of an array starting from zero.

Output:

```

$ g++ point.cpp
$ a.out
21 21 21

```

3. What is the output of this program?

```

1.  #include <iostream>
2.  using namespace std;
3.  int main()
4.  {
5.      int a[2][4] = {3, 6, 9, 12, 15, 18, 21, 24};
6.      cout << *(a[1] + 2) << *((a + 1) + 2) << 2[1[a]];
7.      return 0;
8.  }

```

- a) ava
- b) java
- c) c++
- d) compile time error

View Answer

Answer: a

Explanation: In this program we are moving the pointer from first position to second position and printing the remaining value.

Output:

```

$ g++ point1.cpp
$ a.out
ava

```

3. What is the output of this program?

```
1.    #include <iostream>
2.    using namespace std;
3.    int main()
4.    {
5.        int a[2][4] = {3, 6, 9, 12, 15, 18, 21, 24};
6.        cout << *(a[1] + 2) << (*(a + 1) + 2) << 2[1[a]];
7.        return 0;
8.    }
```

- a) 4
- b) 5
- c) 6
- d) 7

View Answer

Answer: b

Explanation: In this program, we are making the pointer point to next value and printing it.

```
$ g++ point3.cpp
$ a.out
5
```

3. What is the output of this program?

```
1.    #include <iostream>
2.    using namespace std;
3.    int main()
4.    {
5.        int a[2][4] = {3, 6, 9, 12, 15, 18, 21, 24};
6.        cout << *(a[1] + 2) << (*(a + 1) + 2) << 2[1[a]];
7.        return 0;
8.    }
```

- a) 4
- b) 5
- c) address of arr
- d) 7

View Answer

Answer: c

Explanation: As we counted to print only arr, it will print the address of the array.

Output:

advertisement

```
$ g++ point2.cpp
$ a.out
0xbfb1cfff
```

3. What is the output of this program?

```

1.  #include <iostream>
2.  using namespace std;
3.  int main()
4.  {
5.      int a[2][4] = {3, 6, 9, 12, 15, 18, 21, 24};
6.      cout << *(a[1] + 2) << *(* (a + 1) + 2) << 2[1[a]];
7.      return 0;
8.  }
```

a) 10,20,30,40,50,

b) 1020304050

c) compile error

d) runtime error

View Answer

Answer: a

Explanation: In this program, we are just assigning a value to the array and printing it and immediately dereferencing it.

Output:

```

$ g++ point4.cpp
$ a.out
10,20,30,40,50,
```

3. What is the output of this program?

```

1.  #include <iostream>
2.  using namespace std;
3.  int main()
4.  {
5.      int a[2][4] = {3, 6, 9, 12, 15, 18, 21, 24};
6.      cout << *(a[1] + 2) << *(* (a + 1) + 2) << 2[1[a]];
7.      return 0;
8.  }
```

a) 12

b) 5

c) 13

d) error

View Answer

Answer: c

Explanation: In this program, we are adding the value 9 to the initial value of the array, So it's printing as 13.

Output:

```

$ g++ point5.cpp
$ a.out
13
```

C++ Programming Questions and Answers – Constants

1. The constants are also called as

- a) const
- b) preprocessor
- c) literals
- d) none of the mentioned

View Answer

Answer: c

Explanation: Other name for Constants are literals.

2. What are the parts of the literal constants?

- a) integer numerals
- b) floating-point numerals
- c) strings and boolean values
- d) all of the mentioned

View Answer

Answer: d

Explanation: Because these are the types used to declare variables and so these can be declared as constants.

3. How are the constants declared?

- a) const keyword
- b) #define preprocessor
- c) both const keyword and #define preprocessor
- d) none of the mentioned

View Answer

Answer: c

Explanation: The const will declare with a specific type value and #define is used to declare user-defined constants.

- a) 5
- b) 6
- c) Error
- d) None of the mentioned

View Answer

Answer: c

Explanation: We cannot modify a constant integer value.

- a) 12.5664
- b) 13.5664
- c) 10
- d) compile time error

View Answer

Answer: a

Explanation: In this program, we are finding the area of the circle by using concern formula.

Output:

```
$ g++ cons.cpp
$ a.out
12.5664
```

6. Which of the following statement is not true about preprocessor directives?

- a) These are lines read and processed by the preprocessor
- b) They do not produce any code by themselves
- c) These must be written on their own line
- d) They end with a semicolon

View Answer

Answer: d

Explanation: No terminating character required for preprocessor directives statements.

7. Regarding the following statement which of the statements is true?

```
const int a = 100;
```

- a) Declares a variable a with 100 as its initial value
- b) Declares a construction a with 100 as its initial value
- c) Declares a constant a whose value will be 100
- d) Constructs an integer type variable with an as identifier and 100 as the value

View Answer

Answer: c

Explanation: Because the const is used to declare non-changeable values only.

8. The difference between x and 'x' is

- a) The first one refers to a variable whose identifier is x and the second one refers to the character constant x
- b) The first one is a character constant x and the second one is the string literal x
- c) Both are same
- d) None of the mentioned

View Answer

Answer: a

Explanation: In a C++ code, names with quotes like 'x' represent a character or string (in case of collection of characters) whereas without quotes they represent an identifier.

9. How to declare a wide character in the string literal?

- a) L prefix
- b) l prefix
- c) W prefix
- d) none of the mentioned

View Answer

Answer: a

Explanation: It can turn this as the wide character instead of narrow characters.

C++ Programming Questions and Answers – References

1. Which value can we not assign to reference?

- a) integer
- b) floating
- c) unsigned
- d) null

View Answer

Answer: d

Explanation: If it can be assigned with a null value means, it is a copy of the pointer.

2. Identify the incorrect statement

- a) Reference is the alternate name of the object
- b) A reference value once defined can be reassigned
- c) A reference value once defined cannot be reassigned
- d) None of the mentioned

View Answer

Answer: b

Explanation: Reference is a thing which points to the valid memory address, so it can't be redesigned.

3. Which reference modifier is used to define reference variable?

- a) &
- b) \$
- c) #
- d) none of the mentioned

View Answer

Answer: a

Explanation: & aka 'ampersand' used to define a reference variable.

a) In swap 105 In main 105

b) In swap 105 In main 510

c) In swap 510 In main 105

d) None of the mentioned

View Answer

Answer: a

Explanation: As the function is called by reference i.e. all the changes are done directly into the memories of a and b. Therefore changes made to a and b in swap function is reflected back to main function. Hence the values of a and b in swap as well as in main function is changed.

Output:

```
$ g++ ref.cpp
$ a.out
In swap 105 In main 105
```

5. What does a reference provide?

- a) Alternate name for the class
- b) Alternate name for the variable
- c) Alternate name for the pointer
- d) None of the mentioned

View Answer

Answer: b

Explanation: Because we are pointing memory address using the temp variable.

- a) 9
- b) 10
- c) error
- d) 11

View Answer

Answer: b

Explanation: The value is declared and it is incremented, so its value is 10.

```
$ g++ ref1.cpp
$ a.out
10
```

- a) Hello world
- b) Hello
- c) world
- d) compile time error

View Answer

Answer: a

Explanation: In this program we used the concept of constant casting to cast the variable and printing it.

Output:

```
$ g++ ref2.cpp
$ a.out
Hello world
```

8. Identify the correct sentence regarding inequality between reference and pointer.

- a) we can not create the array of reference
- b) we can create the Array of reference
- c) we can use reference to reference
- d) none of the mentioned

View Answer

Answer: a

Explanation: It is not allowed in C++ to make array of references. To test check following array:

```
int &arr[] = {&a, &b, &c};
```

This will give error.

C++ Programming Questions and Answers – References – 2

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
#include <cstdlib>

using namespace std;

int main(int argc, char const *argv[])
{
    int &p;
    int a = 5;
    &p = a;
    cout<<p;
    return 0;
}
```

1. What are the references in C++?

- a) An alternative name for already existing variables
- b) A pointer to a variable
- c) A new type of variables
- d) A new type of constant variable

View Answer

Answer: a

Explanation: References are an alternative name for an already defined variable. They are different from pointers.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
#include <cstdlib>

using namespace std;

int main(int argc, char const *argv[])
{
    int &p;
    int a = 5;
    &p = a;
    cout<<p;
    return 0;
}
```

- a) 5
- b) Run-time error
- c) Segmentation fault
- d) Compile-time error

View Answer

Answer: d

Explanation: References require are other names for variables not for a constant literal. No such assignment are allowed in C++.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
#include <cstdlib>

using namespace std;
```



```
int main(int argc, char const *argv[])
{
    int &p;
    int a = 5;
    &p = a;
    cout<<p;
    return 0;
}
```

- a) 5
- b) 55
- c) Error
- d) Segmentation fault

[View Answer](#)

Answer: c

Explanation: Every reference should be initialized during its declaration but as p is not initialized here therefore the program gives error.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
#include <cstdlib>

using namespace std;

int main(int argc, char const *argv[])
{
    int &p;
    int a = 5;
    &p = a;
    cout<<p;
    return 0;
}
```

- a) 5
- b) Run-time error
- c) Segmentation fault
- d) Compile-time error

[View Answer](#)

Answer: a

Explanation: In this program, every thing is correct so the program runs perfectly and prints the 5 as output.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
#include <cstdlib>

using namespace std;

int main(int argc, char const *argv[])
{
    int &p;
    int a = 5;
    &p = a;
    cout<<p;
    return 0;
}
```

- a) 5
- b) Run-time error
- c) Segmentation fault
- d) Compile-time error

View Answer

Answer: d

Explanation: A pointer cannot be directly assigned to references, because types of pointer(int*) and reference(int) are different here. You need to think before assigning two variable of different types otherwise the program throws error.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
#include <cstdlib>

using namespace std;

int main(int argc, char const *argv[])
{
    int &p;
    int a = 5;
    &p = a;
    cout<<p;
    return 0;
}
```

- a) 5
- b) Address of pointer a
- c) Address of pointer p
- d) Error

View Answer

Answer: b

Explanation: The program is correct so the the program runs perfectly. It is way to assign pointers to references. The program prints the address of a because it is an alias for pointer p and pointer p stores the address of a therefore answer is address of a.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
#include <cstdlib>

using namespace std;

int main(int argc, char const *argv[])
{
    int &p;
    int a = 5;
    &p = a;
    cout<<p;
    return 0;
}
```

7. What is the difference between references and pointers?

- a) References are an alias for a variable whereas pointer stores the address of a variable
- b) References and pointers are similar
- c) References stores address of variables whereas pointer points to variables
- d) Pointers are an alias for a variable whereas references stores the address of a variable

View Answer

Answer: a

Explanation: References are an alias/another name for a variable whereas pointer stores the address of a variable. Pointers need to be de-reference before use whereas references need not.

3. What is the output of the following C++ code?

```
#include <iostream>
```

```
#include <string>
#include <cstdlib>

using namespace std;

int main(int argc, char const *argv[])
{
    int &p;
    int a = 5;
    &p = a;
    cout<<p;
    return 0;
}
```

8. Pick the correct statement about references in C++.

- a) References stores the address of variables
- b) References and variables both have the same address
- c) References use dereferencing operator(*) to access the value of variable its referencing
- d) References were also available in C

View Answer

Answer: b

Explanation: References and variable it is referring to shares the same address. References do not consume extra address. References do not store the address of other variables. No dereferencing operator required while using references. References are not available in C++.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
#include <cstdlib>

using namespace std;

int main(int argc, char const *argv[])
{
    int &p;
    int a = 5;
    &p = a;
    cout<<p;
    return 0;
}
```

- a) Address of a followed by 5 in next line
- b) Address of p followed by 5 in next line
- c) Address of a followed by Address of a in next line
- d) Address of p followed by Address of q in next line

View Answer

Answer: a

Explanation: Pointer p stores the address of variable whereas q is alias for variable a therefore when p is printed it prints the address of a and when q is printed value of a is printed.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
#include <cstdlib>

using namespace std;

int main(int argc, char const *argv[])
{
    int &p;
    int a = 5;
    &p = a;
```

```

        cout<<p;
        return 0;
}

```

- a) Address of a followed by 5 in next line
- b) Address of p followed by 5 in next line
- c) Run time error
- d) Compile time error

View Answer

Answer: d

Explanation: References uses no * operator to access the value of variables it is referring to therefore no program gives error as we are using * operator.

3. What is the output of the following C++ code?

```

#include <iostream>
#include <string>
#include <cstdlib>

using namespace std;

int main(int argc, char const *argv[])
{
    int &p;
    int a = 5;
    &p = a;
    cout<<p;
    return 0;
}

```

- b) Address of p followed by 5 in next line
- c) 5 followed by Address of a in next line
- d) Address of a followed by Address of a in next line

View Answer

Answer: d

Explanation: Both variable and reference shares the same address so the output will be two times the address of a, because references are other name for same variable not a new variable with separate memory.

3. What is the output of the following C++ code?

```

#include <iostream>
#include <string>
#include <cstdlib>

using namespace std;

int main(int argc, char const *argv[])
{
    int &p;
    int a = 5;
    &p = a;
    cout<<p;
    return 0;
}

```

- a) NULL
- b) 0
- c) Address of NULL
- d) Error

View Answer

Answer: d

Explanation: NULL cannot be assigned to references therefore the program gives error. Here it is an int reference and NULL is not an int therefore cannot be assigned to this reference.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
#include <cstdlib>

using namespace std;

int main(int argc, char const *argv[])
{
    int &p;
    int a = 5;
    &p = a;
    cout<<p;
    return 0;
}
```

13. Pick the correct statement about references.

- a) References can be assigned value NULL
- b) References once assigned cannot be changed to refer another variable
- c) Reference should not be initialized when created
- d) Reference is the same as pointers

View Answer

Answer: b

Explanation: References are should be initialized during its creation and once assigned cannot be changed to refer another variable. References cannot be assigned NULL value. References and pointers are two different concepts.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
#include <cstdlib>

using namespace std;

int main(int argc, char const *argv[])
{
    int &p;
    int a = 5;
    &p = a;
    cout<<p;
    return 0;
}
```

14. Which of the following operator is used while declaring references?

- a) *
- b) &
- c) ^
- d) ->

View Answer

Answer: b

Explanation: & operator is used for assigning references.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
#include <cstdlib>
```

```
using namespace std;  
  
int main(int argc, char const *argv[])  
{  
    int &p;  
    int a = 5;  
    &p = a;  
    cout<<p;  
    return 0;  
}
```

- a) 5
- b) 10
- c) Error
- d) Segmentation fault

View Answer

Answer: c

Explanation: As we are passing a as const reference to function therefore its value cannot be changes inside the function. So the program gives error.

ayzom.com

C++ Programming Questions and Answers – References – 3

a) 343

b) 336

c) 120

d) 840

View Answer

Answer: a

Explanation: In this program as one parameter is passed by value and other is passed by reference so after 4 calls when $c == 0$, then the value of $x = 7$ and as x is passed by reference so all the changes will be reflected back in all the previous calls hence the answer $1*7*7*7 = 343$.

2. Which of the following is incorrect?

a) References cannot be NULL

b) A reference must be initialized when declared

c) Once a reference is declared, it cannot be modified later to reference another object i.e. it cannot be reset

d) References cannot refer to a constant value

View Answer

Answer: d

Explanation: C++ allows references to refer to a constant value by making constant references. For example:

```
const int a = 5;
```

```
const int &ref = a;
```

is an example of that.

3. Which of the following function must use reference.

a) Assignment operator function

b) Copy Constructor

c) Destructor

d) Parameterized constructor

View Answer

Answer: b

Explanation: We don't need references in case of assignment, destructor or constructor. But in case of a copy constructor, we need to call copy constructor because if we use pass by value then as copy constructor itself is a function. So if we pass an argument by pass by value method in a copy constructor, a call to copy constructor would be made to again call copy constructor which becomes an endless chain of calls. Therefore compiler doesn't allow parameters to be passed by value.

a) 30

b) 10

c) Error

d) Segmentation fault

View Answer

Answer: a

Explanation: A function returning value by reference can be used as lvalue i.e. it can be used on the left side of an expression. Here when we are doing `fun() = 30` then we are changing the value of x (i.e. value returning) to and as x is static therefore it will not be initialized again so the value of x becomes 30 hence the output is 30.

a) 30

b) 10

c) Error

d) Segmentation fault

View Answer

Answer: d

Explanation: In this case we are trying to assign 30 to a local variable which is returned from the function func() which will be destroyed after the function call hence next time this assignment is not correct hence segmentation fault.

View Answer

Answer: a

Explanation: As we know references are alias for a variable so the value of a variable can be changed using alias hence both ref and x are same therefore changing the value of one effects the value of other.

7. How a reference is different from a pointer?

- a) A reference cannot be null
- b) A reference once established cannot be changed
- c) The reference doesn't need an explicit dereferencing mechanism
- d) All of the mentioned

View Answer

Answer: d

Explanation: References can never be NULL. It is not allowed to change a reference once allocated. Referencing does not need an explicit referencing operator.

8. Which of the following statement(s) is/are correct?

- a) * operator is used to declare a reference
- b) A reference variable defined to refer a particular variable can refer to any other variable also
- c) References must always be initialized inside classes
- d) A variable can have more than one references

View Answer

Answer: d

Explanation: A variable can have multiple references as references are nothing just another name for a variable hence a variable can have more than one references.

C++ Programming Questions and Answers – Pointer to Void

1. The void pointer can point to which type of objects?

- a) int
- b) float
- c) double
- d) all of the mentioned

View Answer

Answer: d

Explanation: Because it doesn't know the type of object it is pointing to, So it can point to all objects.

2. When does the void pointer can be dereferenced?

- a) when it doesn't point to any value
- b) when it cast to another type of object
- c) using delete keyword
- d) none of the mentioned

View Answer

Answer: b

Explanation: By casting the pointer to another data type, it can be dereferenced from the void pointer.

3. The pointer can point to any variable that is not declared with which of these?

- a) const
- b) volatile
- c) both const & volatile
- d) static

View Answer

Answer: c

Explanation: None.

4. A void pointer cannot point to which of these?

- a) methods in c++
- b) class member in c++
- c) all of the mentioned
- d) none of the mentioned

View Answer

Answer: d

Explanation: None.

a) abcdefghij

b) address of string "abcdefghij"

c) compile time error

d) runtime error

View Answer

Answer: b

Explanation: Even though it is a void pointer, we gets the address.

Output:

```
$ g++ b.cpp
$ a.out
0x8048714
```

- a) equal
- b) no output
- c) compile error
- d) runtime error

View Answer

Answer: a

Explanation: The void pointer is easily converted to any other type of pointer, so these are equal.

Output:

```
$ g++ poi4.cpp
$ a.out
equal
```

- a) 2d
- b) two memory addresses
- c) both of the mentioned
- d) none of the mentioned

View Answer

Answer: b

Explanation: Because the data points to the address value of the variables only, So it is printing the memory address of these two variable.

Output:

```
$ g++ poi2.cpp
$ a.out
the data points to the integer value0xbfc81824 the data now points to the character0xbfc8182f
```

- a) 5
- b) 6
- c) compile time error
- d) runtime error

View Answer

Answer: a

Explanation: We just casted this from void to int, so it prints 5

Output:

```
$ g++ poi1.cpp
$ a.out
5
```

- a) 8, memory address
- b) 8.14
- c) memory address
- d) none of the mentioned

View Answer

Answer: a

Explanation: In this program, we are just adding the two values and printing it.

Output:

```
$ g++ poi.cpp
$ a.out
8
0xbfef0378
```

10. What we can't do on a void pointer?

- a) pointer arithmetic

- b) pointer functions
 - c) both of the mentioned
 - d) none of the mentioned
- View Answer

Answer: a

Explanation: Because the void pointer is used to cast the variables only, So pointer arithmetic can't be done in a void pointer.

ayzom.com

C++ Programming Questions and Answers – Structures

1. The data elements in the structure are also known as what?

- a) objects
- b) members
- c) data
- d) none of the mentioned

View Answer

Answer: b

Explanation: Variables declared inside a class are called as data elements or data members.

2. What will be used when terminating a structure?

- a) :
- b) }
- c) ;
- d) ::

View Answer

Answer: c

Explanation: While terminating a structure, a semicolon is used to end this up.

3. What will happen when the structure is declared?

- a) it will not allocate any memory
- b) it will allocate the memory
- c) it will be declared and initialized
- d) none of the mentioned

View Answer

Answer: a

Explanation: While the structure is declared, it will not be initialized, So it will not allocate any memory.

4. The declaration of the structure is also called as?

- a) structure creator
- b) structure signifier
- c) structure specifier
- d) none of the mentioned

View Answer

Answer: c

Explanation: The structure declaration with open and close braces and with a semicolon is also called structure specifier.

- a) 123
john
- b) john
john
- c) compile time error
- d) none of the mentioned

View Answer

Answer: a

Explanation: We are copying the value john to the name and then we are printing the values that are in the program.

Output:

```
$ g++ stu.cpp
$ a.out
```

123
john

- a) 19845
 - b) 20000
 - c) 15000
 - d) 19844
- [View Answer](#)

Answer: a

Explanation: In this program, we are just converting the given hours and minutes into seconds.

Output:

```
$ g++ stu1.cpp
$ a.out
Total seconds:19845
```

- a) Adidas \$ 9.99Adidas \$ 1.11
 - b) Adidas \$ 9.99Adidas \$ 9.11
 - c) Adidas \$ 9.99Adidas \$ 11.11
 - d) none of the mentioned
- [View Answer](#)

Answer: a

Explanation: We copied the value of shoe1 into shoe2 and divide the shoe2 value by 9, So this is the output.

Output:

```
$ g++ stu2.cpp
$ a.out
Adidas $ 9.99
Adidas $ 1.11
```

- a) 252
 - b) 253
 - c) 254
 - d) 262
- [View Answer](#)

Answer: a

Explanation: In this program, We are dividing the values of a and b, printing it.

Output:

```
$ g++ stu5.cpp
$ a.out
252
```

9. Which of the following is a properly defined structure?

- a) struct {int a;}
- b) struct a_struct {int a;}
- c) struct a_struct int a;
- d) struct a_struct {int a;};

[View Answer](#)

Answer: d

Explanation: option struct {int a;} is not correct because name of structure and ;(after declaration) are missing. In option struct a_struct {int a;} ; is missing. In option struct a_struct int a; {} are missing.

10. Which of the following accesses a variable in structure *b?

- a) b->var;
- b) b.var;
- c) b-var;
- d) b>var;

View Answer

Answer: a

Explanation: Because arrow operator(->) is used to access members of structure pointer whereas dot operator(.) is used to access normal structure variables.

ayzom.com

C++ Programming Questions and Answers – Character Classification

1. Which function is used to check whether a character is an alphabet?

- a) isalpha()
- b) isalnum()
- c) isdigit()
- d) isblank()

View Answer

Answer: a

Explanation: Character classification provides isalpha() function to check whether a character in C++ is an alphabet or not.

2. Which function is used to check whether a character is an alphabet or number?

- a) isalpha()
- b) isalnum()
- c) isdigit()
- d) isblank()

View Answer

Answer: b

Explanation: Character classification provides isalnum() function to check whether a character in C++ is alphabet or number.

3. Which function is used to check whether a character is a number?

- a) isalpha()
- b) isalnum()
- c) isdigit()
- d) isblank()

View Answer

Answer: c

Explanation: Character classification provides isdigit() function to check whether a character in C++ is number or not.

4. Which function is used to check whether a character is a tab or space?

- a) isalpha()
- b) isalnum()
- c) isdigit()
- d) isblank()

View Answer

Answer: d

Explanation: Character classification provides isblank() function to check whether a character in C++ is space or tab.

5. Which function is used to check whether a character is tab or space or whitespace control code(\n,\r,etc.)?

- a) isspace()
- b) isalnum()
- c) iscntrl()
- d) isblank()

View Answer

Answer: a

Explanation: Character classification provides isspace() function to check whether a character in C++ is tab or space or whitespace control code(\n, \r, etc.).

6. Which function is used to check whether a character is tab or a control code?

- a) isspace()

b) isalnum()
c) iscntrl()
d) isblank()
View Answer

Answer: c

Explanation: Character classification provides iscntrl() function to check whether a character in C++ is tab or a control code.

7. Which function is used to check whether a character is printable on console?

a) isxdigit()
b) isprint()
c) iscntrl()
d) ispunct()
View Answer

Answer: b

Explanation: Character classification provides isprint() function to check whether a character in C++ is printable on console.

8. Which function is used to check whether a character is hexadecimal?

a) isxdigit()
b) isprint()
c) iscntrl()
d) ispunct()
View Answer

Answer: a

Explanation: Character classification provides isxdigit() function to check whether a character in C++ is hexadecimal.

9. Which function is used to check whether a character is punctuation mark?

a) isxdigit()
b) isprint()
c) iscntrl()
d) ispunct()
View Answer

Answer: d

Explanation: Character classification provides ispunct() function to check whether a character in C++ is punctuation mark.

a) 111110111110
b) 111111111110
c) 111000111110
d) 111110000000
View Answer

Answer: a

Explanation: In this program we are checking whether a character is an alphabet or not so in "Hello World" except space everything is alphabet, therefore, we have 11111011111 but it is followed by a 0 because every string is followed by a null character which is not alphabet, therefore, we have 0 at the end of the binary string.

View Answer

Answer: d

Explanation: In this program, we are first checking the alphabets in the string then digits in the string so accordingly one can find the answer.

a) 111000111110
b) 111111111110
c) 111110111110
d) 111110000000

View Answer

Answer: c

Explanation: In this program we are checking the presence of alphabets and digits in the string so accordingly one can find the answer.

- a) 11111111110
- b) 000001000001
- c) 111000111110
- d) 111110000000

View Answer

Answer: b

Explanation: In this program we are checking the presence of control codes i.e. \n, \r, \r\n, \t, etc. in the string so accordingly one can find the answer.

- a) 10000010
- b) 111111111110
- c) 111000111110
- d) 111110000000

View Answer

Answer: a

Explanation: In this program we are checking the presence of quotes(' ', etc.) in the string so accordingly one can find the answer.

- a) 111001100011110000000111100
- b) 101010101010101001010101010
- c) 111111000000000000000000000
- d) 11111111000001111011110111

View Answer

Answer: c

Explanation: In this program, we are checking the presence of hexadecimal characters in the string and as only a, b, c, d, e and f are used as hexadecimal characters therefore only first bits are 1 and others are 0.

C++ Programming Questions and Answers – Operators

1. Which operator is having the right to left associativity in the following?

- a) Array subscripting
- b) Function call
- c) Addition and subtraction
- d) Type cast

View Answer

Answer: d

Explanation: There are many right to left associativity operators in C++, which means their evaluation is done from right to left. Type Cast is one of them. Here is a link of associativity of operators: <https://github.com/MicrosoftDocs/cpp-docs/blob/master/docs/cpp/cpp-built-in-operators-precedence-and-associativity.md>

2. Which operator is having the highest precedence?

- a) postfix
- b) unary
- c) shift
- d) equality

View Answer

Answer: a

Explanation: The operator which is having highest precedence is postfix and lowest is equality.

3. What is this operator called ??

- a) conditional
- b) relational
- c) casting operator
- d) none of the mentioned

View Answer

Answer: a

Explanation: In this operator, if the condition is true means, it will return the first operator, otherwise second operator.

- a) 35
- b) 20
- c) 25
- d) 30

View Answer

Answer: b

Explanation: Because the * operator is having highest precedence, So it is executed first and then the + operator will be executed.

Output:

```
$ g++ op1.cpp
$ a.out
20
```

5. What is the use of dynamic_cast operator?

- a) it converts virtual base class to derived class
- b) it converts the virtual base object to derived objects
- c) it will convert the operator based on precedence
- d) none of the mentioned

View Answer

Answer: a

Explanation: Because the `dynamic_cast` operator is used to convert from base class to derived class.

- a) 5 6
- b) 6 5
- c) 6 7
- d) none of the mentioned

View Answer

Answer: a

Explanation: It is a separator here. In C, the value a is stored in c and in d the value b is stored in d because of the bracket.

Output:

```
$ g++ op3.cpp
$ a.out
5      6
```

- a) 1000
- b) 11
- c) 1010
- d) 1001

View Answer

Answer: c

Explanation: j starts with the value 10. j is then incremented to 11. Next, j is added to 100. Finally, j (still containing 11) is added to 999 which yields the result 1010.

Output:

```
$ g++ op2.cpp
$ a.out
1010
```

- a) 749735
- b) 736749
- c) 367497
- d) none of the mentioned

View Answer

Answer: a

Explanation: Because of the precedence the pre-increment and post increment operator, we got the output as 749736.

Output:

```
$ g++ op.cpp
$ a.out
749735
```

- a) 6
- b) 5
- c) 4
- d) 7

View Answer

Answer: a

Explanation: Here the condition is false on conditional operator, so the b value is assigned to c.

Output:

```
$ g++ op1.cpp
$ a.out
6
```

- a) 20 10
- b) 10 21
- c) 21 10
- d) none of the mentioned

View Answer

Answer: c

Explanation: In this program, we are casting the operator to integer, So it is printing as 21 and 10.

Output:

```
$ g++ op5.cpp
$ a.out
21      10
```

ayzom.com

C++ Programming Questions and Answers – Statements

1. How are many sequences of statements present in c++?

- a) 4
- b) 3
- c) 5
- d) 6

View Answer

Answer: c

Explanation: There are five sequences of statements. They are Preprocessor directives, Comments, Declarations, Function Declarations, Executable statements.

2. The if..else statement can be replaced by which operator?

- a) Bitwise operator
- b) Conditional operator
- c) Multiplicative operator
- d) None of the mentioned

View Answer

Answer: b

Explanation: In the conditional operator, it will predicate the output using the given condition.

3. The switch statement is also called as?

- a) choosing structure
- b) selective structure
- c) certain structure
- d) none of the mentioned

View Answer

Answer: b

Explanation: The switch statement is used to choose the certain code to execute, So it is also called as selective structure.

4. The destination statement for the goto label is identified by what label?

- a) \$
- b) @
- c) *
- d) :

View Answer

Answer: d

Explanation: : colon is used at the end of labels of goto statements.

a) 543

b) 54

c) 5432

d) 53

View Answer

Answer: a

Explanation: In this program, We are printing the numbers in reverse order but by using break statement we stopped printing on 3.

Output:

```
$ g++ stat.cpp
$ a.out
543
```

- a) 1010
 - b) 10
 - c) infinitely print 10
 - d) compile time error
- View Answer

Answer: d

Explanation: Because the break statement need to be presented inside a loop or a switch statement.

- a) error
 - b) 15
 - c) infinite times of printing n
 - d) none of the mentioned
- View Answer

Answer: c

Explanation: There is not a condition in the for loop, So it will loop continuously.

- a) 0123456789
 - b) 10
 - c) 012345678910
 - d) compile time error
- View Answer

Answer: b

Explanation: for loop with a semicolon is called as body less for loop. It is used only for incrementing the variable values. So in this program the value is incremented and printed as 10.

Output:

```
$ g++ stat2.cpp
$ a.out
10
```

9. How many types of loops are there in C++?

- a) 4
- b) 2
- c) 3
- d) 1

View Answer

Answer: a

Explanation: There are four types of loop. They are the while, do while, nested, for the loop.

10. Which looping process is best used when the number of iterations is known?

- a) for
- b) while
- c) do-while
- d) all looping processes require that the iterations be known

View Answer

Answer: a

Explanation: Because in for loop we are allowed to provide starting and ending conditions of loops, hence fixing the number of iterations of loops, whereas no such things are provided by other loops.

C++ Programming Questions and Answers – Comments and Indentation

1. How many types of comments are there in c++?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: b

Explanation: There are two types of comments in C++. Single line comments uses double slash //. Multiple line comments uses /* comment inside */.

2. What is a comment in c++?

- a) comments are parts of the source code disregarded by the compiler
- b) comments are executed by the compiler to find the meaning of the comment
- c) comments are executable
- d) none of the mentioned

View Answer

Answer: a

Explanation: Comments are used to add meaning to the program.

3. What type of comments does c++ support?

- a) single line
- b) multiline
- c) single line and multi-line
- d) none of the mentioned

View Answer

Answer: c

Explanation: C++ provides two types of comments in programs. They are single line(using //) or multiple line (using /*..... */) comments.

- a) hello world
- b) hello
- c) compile time error
- d) none of the mentioned

View Answer

Answer: c

Explanation: Because the slash should need to be forward not backward.

5. What is used to write multi line comment in c++?

- a) /* */
- b) /\$ \$/
- c) //
- d) none of the mentioned

View Answer

Answer: a

Explanation: The /* is used to write the multi line comment.

6. What is the use of the indentation in c++?

- a) distinguishes between comments and code
- b) r distinguishes between comments and outer data

- c) all of the mentioned
- d) none of the mentioned

View Answer

Answer: a

Explanation: To distinguish between different parts of the program like comments, codes, etc.

- a) 6
- b) 24
- c) segmentation fault
- d) compile time error

View Answer

Answer: c

Explanation: As we have given in the function as $a+1$, it will exceed the size and so it arises the segmentation fault.

Output:

```
$ g++ arg3.cpp
$ a.out
segmentation fault
```

- a) 100
- b) compile time error
- c) 144
- d) 110

View Answer

Answer: d

Explanation: We have increased the x value in operand as $x + 1$, so it will return as 110.

Output:

```
$ g++ arg2.cpp
$ a.out
110
```

- a) 11
- b) 12
- c) 13
- d) compile time error

View Answer

Answer: c

Explanation: The value of a has been changed to 7, So it returns as 13.

Output:

```
$ g++ arg1.cpp
$ a.out
13
```

10. What will happen when we use void in argument passing?

- a) It will not return value to its caller
- b) It will return value to its caller
- c) Maybe or may not be return value to its caller
- d) None of the mentioned

View Answer

Answer: a

Explanation: As void is not having any return value, it will not return the value to the caller.

- a) 2 3
- b) 6 9
- c) 2 15
- d) compile time error

View Answer

Answer: c

Explanation: We have passed three values and it will manipulate according to the given condition and yield the result as 2 15.

Output:

```
$ g++ arg.cpp
$ a.out
2 15
```

ayzom.com

C++ Programming Questions and Answers – Function Declarations

1. Where does the execution of the program starts?

- a) user-defined function
- b) main function
- c) void function
- d) none of the mentioned

View Answer

Answer: b

Explanation: Normally the execution of the program in c++ starts from main only.

2. What are mandatory parts in the function declaration?

- a) return type, function name
- b) return type, function name, parameters
- c) parameters, function name
- d) none of the mentioned

View Answer

Answer: a

Explanation: In a function, return type and function name are mandatory all else are just used as a choice.

3. which of the following is used to terminate the function declaration?

- a) :
- b))
- c) ;
- d) none of the mentioned

View Answer

Answer: c

Explanation: ; semicolon is used to terminate a function declaration statement in C++.

4. How many can max number of arguments present in function in the c99 compiler?

- a) 99
- b) 90
- c) 102
- d) 127

View Answer

Answer: d

Explanation: C99 allows to pass a maximum of 127 arguments in a function.

5. Which is more effective while calling the functions?

- a) call by value
- b) call by reference
- c) call by pointer
- d) none of the mentioned

View Answer

Answer: b

Explanation: In the call by reference, it will just passes the reference of the memory addresses of passed values rather than copying the value to new memories which reduces the overall time and memory use.

- a) hai
- b) haihai

- c) compile time error
- d) none of the mentioned

View Answer

Answer: c

Explanation: We have to use the semicolon to declare the function in line 3. This is called a function declaration and a function declaration ends with a semicolon.

- a) 10
- b) 20
- c) compile time error
- d) none of the mentioned

View Answer

Answer: a

Explanation: In this program, we called by value so the value will not be changed, So the output is 10

Output:

advertisement

```
$ g++ fun.cpp  
$ a.out  
10
```

8. What is the scope of the variable declared in the user defined function?

- a) whole program
- b) only inside the {} block
- c) the main function
- d) none of the mentioned

View Answer

Answer: b

Explanation: The variable is valid only in the function block as in other.

9. How many minimum number of functions should be present in a C++ program for its execution?

- a) 0
- b) 1
- c) 2
- d) 3

View Answer

Answer: b

Explanation: The execution of a C++ program starts from main function hence we require atleast 1 function to be present in a C++ program to execute and i.e. main function.

C++ Programming Questions and Answers – Functions

1. Which of the following is the default return value of functions in C++?

- a) int
- b) char
- c) float
- d) void

View Answer

Answer: a

Explanation: C++ uses int as the default return values for functions. It also restricts that the return type of the main function must be int.

2. What happens to a function defined inside a class without any complex operations (like looping, a large number of lines, etc)?

- a) It becomes a virtual function of the class
- b) It becomes a default calling function of the class
- c) It becomes an inline function of the class
- d) The program gives an error

View Answer

Answer: c

EXplanation: Any function which is defined inside a class and has no complex operations like loops, a large number of lines then it is made inline.

3. What is an inline function?

- a) A function that is expanded at each call during execution
- b) A function that is called during compile time
- c) A function that is not checked for syntax errors
- d) A function that is not checked for semantic analysis

View Answer

Answer: a

Explanation: Inline function is those which are expanded at each call during the execution of the program to reduce the cost of jumping during execution.

4. An inline function is expanded during _____

- a) compile-time
- b) run-time
- c) never expanded
- d) end of the program

View Answer

Answer: b

Explanation: An inline function is expanded during the runtime of a program.

5. In which of the following cases inline functions may not work?

- i) If the function has static variables.
 - ii) If the function has global and register variables.
 - iii) If the function contains loops
 - iv) If the function is recursive
- a) i, iv
 - b) iii, iv
 - c) ii, iii, iv
 - d) i, iii, iv

View Answer

Answer: d

Explanation: A function is not inline if it has static variables, loops or the function is having any recursive calls.

6. When we define the default values for a function?

- a) When a function is defined
- b) When a function is declared
- c) When the scope of the function is over
- d) When a function is called

View Answer

Answer: b

Explanation: Default values for a function is defined when the function is declared inside a program.

7. Where should default parameters appear in a function prototype?

- a) To the rightmost side of the parameter list
- b) To the leftmost side of the parameter list
- c) Anywhere inside the parameter list
- d) Middle of the parameter list

View Answer

Answer: a

Explanation: Default parameters are defined to the rightmost side of parameter list in a function to differentiate between the normal and default parameters for example if a function is defined as `fun(int x = 5, int y)` then if we call `fun(10)` then 10 should be given to x or y because one can apply both logics like `x = 10` already defined and 10 passed is for y but if compiler reads it from left to right it will think it is for x and no parameter is given for y, therefore, the compiler will give error.

8. If an argument from the parameter list of a function is defined constant then _____

- a) It can be modified inside the function
- b) It cannot be modified inside the function
- c) Error occurs
- d) Segmentation fault

View Answer

Answer: b

Explanation: A function is not allowed a constant member of the parameter list.

9. Which of the following feature is used in function overloading and function with default argument?

- a) Encapsulation
- b) Polymorphism
- c) Abstraction
- d) Modularity

View Answer

Answer: c

Explanation: Both of the above types allows a function overloading which is the basic concept of Polymorphism.

- a) 10
- b) 0
- c) Error
- d) Segmentation fault

View Answer

Answer: a

Explanation: Default arguments should always be declared at the rightmost side of the parameter list but the above function has a normal variable at the rightmost side which is a syntax error, therefore the function gives an error.

View Answer

Answer: a

Explanation: As the object declared are of two types one is normal object and other is constant object So normal objects calls normal fun() whereas constant objects calls constant fun().

- a) -5
- b) 0
- c) 10
- d) 5

View Answer

Answer: d

Explanation: C++ allows to define such prototype of the function in which you are not required to give variable names only the default values. While in function definition you can provide the variable names corresponding to each parameter.

- a) 870
- b) 30
- c) Error
- d) Segmentation fault

View Answer

Answer: a

Explanation: As we are passing value by reference therefore the change in the value is reflected back to the passed variable number hence value of number is changed to 870.

14. From which function the execution of a C++ program starts?

- a) start() function
- b) main() function
- c) new() function
- d) all of the mentioned

View Answer

Answer: b

Explanation: The execution of a C++ program starts from the main() function.

15. Which of the following is important in a function?

- a) Return type
- b) Function name
- c) Both return type and function name
- d) The return type, function name and parameter list

View Answer

Answer: c

Explanation: The important things required in a function is its return type and its name other than that parameter list are optional which a function may or may not have.

C++ Programming Questions and Answers – Argument Passing

1. How many ways of passing a parameter are there in c++?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: c

Explanation: There are three ways of passing a parameter. They are pass by value, pass by reference and pass by pointer.

2. Which is used to keep the call by reference value as intact?

- a) static
- b) const
- c) absolute
- d) none of the mentioned

View Answer

Answer: b

Explanation: Because const will not change the value of the variables during the execution.

3. By default how the value are passed in c++?

- a) call by value
- b) call by reference
- c) call by pointer
- d) none of the mentioned

View Answer

Answer: a

Explanation: None.

a) 2 5 10

b) 2 4 5

c) 2 6 14

d) none of the mentioned

View Answer

Answer: c

Explanation: Because we multiplied the values by 2 in the copy function.

Output:

```
$ g++ arg6.cpp
$ a.out
x = 2, y = 6, z = 14
```

a) 10

b) 20

c) 15

d) none of the mentioned

View Answer

Answer: b

Explanation: As the parameter is passed by reference, the value in the original memory of x is changed hence the output is printed as 20.

Output:

```
$ g++ arg5.cpp  
$ a.out  
20
```

- a) 6
- b) 24
- c) segmentation fault
- d) compile time error

View Answer

Answer: c

Explanation: As we have given in the function as $a+1$, it will exceed the size and so it arises the segmentation fault.

Output:

```
$ g++ arg3.cpp  
$ a.out  
segmentation fault
```

- a) 100
- b) compile time error
- c) 144
- d) 110

View Answer

Answer: d

Explanation: We have increased the x value in operand as $x+1$, so it will return as 110.

Output:

```
$ g++ arg2.cpp  
$ a.out  
110
```

- a) 11
- b) 12
- c) 13
- d) compile time error

View Answer

Answer: c

Explanation: The value of a has been changed to 7, So it returns as 13.

Output:

```
$ g++ arg1.cpp  
$ a.out  
13
```

9. What will happen when we use void in argument passing?

- a) It will not return value to its caller
- b) It will return value to its caller
- c) Maybe or may not be return any value to its caller
- d) None of the mentioned

View Answer

Answer: a

Explanation: As void is not having any return value, it will not return the value to the caller.

- a) 2 3
- b) 6 9
- c) 2 15
- d) compile time error

View Answer

Answer: c

Explanation: We have passed three values and it will manipulate according to the given condition and yield the result as 2 15

Output:

```
$ g++ arg.cpp
$ a.out
2 15
```

ayzom.com

C++ Programming Questions and Answers – Value Return

1. How many types of returning values are present in c++?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: c

Explanation: The three types of returning values are return by value, return by reference and return by address.

2. What will you use if you are not intended to get a return value?

- a) static
- b) const
- c) volatile
- d) void

View Answer

Answer: d

Explanation: Void is used to not to return anything.

3. Where does the return statement returns the execution of the program?

- a) main function
- b) caller function
- c) same function
- d) none of the mentioned

View Answer

Answer: b

Explanation: The execution of the program is returned to the point from where the function was called and the function from which this function was called is known as caller function.

- a) 5
- b) 7
- c) either 5 or 7
- d) none of the mentioned

View Answer

Answer: b

Explanation: In this program, we are returning the maximum value by using conditional operator.

Output:

```
$ g++ ret.cpp
$ a.out
7
```

- a) 46.5
- b) 6.50
- c) compile time error
- d) none of the mentioned

View Answer

Answer: a

Explanation: We are returning the value what we get as input.

Output:

```
$ g++ ret1.cpp
$ a.out
46.5
```

- a) 20
- b) 25
- c) 30
- d) 35

[View Answer](#)

Answer: b

Explanation: We are multiplying these values by adding every values.

Output:

```
$ g++ ret.cpp
$ a.out
25
```

7. When will we use the function overloading?

- a) same function name but different number of arguments
- b) different function name but same number of arguments
- c) same function name but same number of arguments
- d) different function name but different number of arguments

[View Answer](#)

Answer: a

Explanation: We use function overloading when we want the same name function to perform different procedure for different types of parameters or different number of parameters provided to the function.

- a) 15
- b) 25
- c) 375
- d) 5

[View Answer](#)

Answer: d

Explanation: In this program, we are finding the gcd of the number.

Output:

```
$ g++ ret5.cpp
$ a.out
5
```

C++ Programming Questions and Answers – Overloaded Function Names

1. Which of the following permits function overloading on c++?

- a) type
- b) number of arguments
- c) type & number of arguments
- d) none of the mentioned

View Answer

Answer: c

Explanation: Both type and number of arguments permits function overloading in C++, like

```
int func(int);
```

```
float func(float, float)
```

Here both type and number of arguments are different.

2. In which of the following we cannot overload the function?

- a) return function
- b) caller
- c) called function
- d) none of the mentioned

View Answer

Answer: a

Explanation: While overloading the return function, it will rise a error, So we can't overload the return function.

3. Function overloading is also similar to which of the following?

- a) operator overloading
- b) constructor overloading
- c) destructor overloading
- d) none of the mentioned

View Answer

Answer: b

Explanation: In constructor overloading, we will be using the same options available in function overloading.

- a) 5500.263
- b) 500.2635
- c) 500.263
- d) none of the mentioned

View Answer

Answer: a

Explanation: In this program, we are printing the values and the values will be print(5) will be printed first because of the order of the execution.

Output:

```
$ g++ over.cpp
$ a.out
5500.263
```

- a) 11 12.1
- b) 12.1 11
- c) 11 12
- d) compile time error

View Answer

Answer: d

Explanation: As one can observe that no function has declaration similar to that of called Add(int, int) and Add(double, double) functions. Therefore, error occurs.

- a) 10.0 5.0
 - b) 5.0 2.5
 - c) 10.0 5
 - d) 10 2.5
- View Answer

Answer: d

Explanation: In this program, we are divide and multiply the values.

Output:

advertisement

```
$ g++ over3.cpp
$ a.out
10      2.5
```

7. Overloaded functions are

- a) Very long functions that can hardly run
- b) One function containing another one or more functions inside it
- c) Two or more functions with the same name but different number of parameters or type
- d) None of the mentioned

View Answer

Answer: c

Explanation: This is the definition of function overloading i.e. function having same name but different number of parameters and types.

8. What will happen while using pass by reference

- a) The values of those variables are passed to the function so that it can manipulate them
- b) The location of variable in memory is passed to the function so that it can use the same memory area for its processing
- c) The function declaration should contain ampersand (& in its type declaration)
- d) All of the mentioned

View Answer

Answer: b

Explanation: In pass by reference, we can use the function to access the variable and it can modify it. Therefore we are using pass by reference.

9. What should be passed in parameters when function does not require any parameters?

- a) void
- b) blank space
- c) both void & blank space
- d) none of the mentioned

View Answer

Answer: b

Explanation: When we does not want to pass any argument to a function then we leave the parameters blank i.e. func() – function without any parameter.

10. What are the advantages of passing arguments by reference?

- a) Changes to parameter values within the function also affect the original arguments
- b) There is need to copy parameter values (i.e. less memory used)
- c) There is no need to call constructors for parameters (i.e. faster)
- d) All of the mentioned

View Answer

Answer: d

Explanation: All above mentioned are advantages and properties of call by reference.

ayzom.com

C++ Programming Questions and Answers – Default Arguments

1. If the user did not supply the value, what value will it take?

- a) default value
- b) rise an error
- c) both default value & rise an error
- d) none of the mentioned

View Answer

Answer: a

Explanation: If the user did not supply the value means, the compiler will take the given value in the argument list.

2. Where can the default parameter be placed by the user?

- a) leftmost
- b) rightmost
- c) both leftmost & rightmost
- d) none of the mentioned

View Answer

Answer: b

Explanation: To avoid the ambiguity between the non-default parameters and default parameters.

3. Which value will it take when both user and default values are given?

- a) user value
- b) default value
- c) custom value
- d) none of the mentioned

View Answer

Answer: a

Explanation: The default value will be used when the user value is not given, So in this case, the user value will be taken.

- a) Flag is true. a = 200
- b) Flag is false. a = 100
- c) Flag is false. a = 200
- d) Flag is true. a = 100

View Answer

Answer: c

Explanation: In this program, we are passing the value, as it evaluates to false, it produces the output as following.

Output:

```
$ g++ def.cpp
$ a.out
Flag is false. a = 200
```

- a) 5
- b) 6
- c) the number you entered
- d) compile time error

View Answer

Answer: c

Explanation: In this program, we are getting a number and printing it.

Output:

```
$ g++ def1.cpp
$ a.out
Please enter a number:
5
Here is your number:5
```

- a) 1st value: 1
10
3
4
b) 1st value: 1
10
3
10
c) compile time error
d) none of the mentioned
View Answer

Answer: a

Explanation: In this program, We are passing the values as by default values rules it is working.

Output:

```
$ g++ def2.cpp
$ a.out
1st value: 1
2nd value: 10
1st value: 3
2nd value: 4
```

7. What we can't place followed by the non-default arguments?

- a) trailing arguments
b) default arguments
c) both trailing & default arguments
d) none of the mentioned
View Answer

Answer: b

Explanation: To avoid the ambiguity in arguments.

eg. if func(int a=3, int b);

so if we call func(5), here will 5 will be value of a or b, because 5 is first parameter so a should be 5 but as only one argument is given b should be 5. So to remove such ambiguity default parameters are kept at the end or rightmost side.

8. If we start our function call with default arguments means, what will be proceeding arguments?

- a) user argument
b) empty arguments
c) default arguments
d) none of the mentioned
View Answer

Answer: c

Explanation: As a rule, the default argument must be followed by default arguments only.

9. What is the default return type of a function?

- a) int
b) void
c) float
d) char
View Answer

Answer: b

Explanation: void is the default return value of any function, to handle both empty and non-empty values.

- a) 15
- b) 10
- c) compile time error
- d) none of the mentioned

View Answer

Answer: c

Explanation: In function parameters the default arguments should always be the rightmost parameters.

ayzom.com

C++ Programming Questions and Answers – Unspecified Number of Arguments

3. What is the output of this program?

```
1.    #include <iostream>
2.    #include <stdarg.h>
3.    using namespace std;
4.    float avg( int Count, ... )
5.    {
6.        va_list Numbers;
7.        va_start(Numbers, Count);
8.        int Sum = 0;
9.        for (int i = 0; i < Count; ++i )
10.           Sum += va_arg(Numbers, int);
11.        va_end(Numbers);
12.        return (Sum/Count);
13.    }
14.    int main()
15.    {
16.        float Average = avg(10, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9);
17.        cout << "Average of first 10 whole numbers : " << Average;
18.        return 0;
19.    }
```

1. Which header file is used to pass unknown number of arguments to function?

- a) stdlib.h
- b) string.h
- c) stdarg.h
- d) none of the mentioned

View Answer

Answer: c

Explanation: Because the cstdlib defines this header file to process the unknown number of arguments.

3. What is the output of this program?

```
1.    #include <iostream>
2.    #include <stdarg.h>
3.    using namespace std;
4.    float avg( int Count, ... )
5.    {
```

```
6.      va_list Numbers;
7.      va_start(Numbers, Count);
8.      int Sum = 0;
9.      for (int i = 0; i < Count; ++i )
10.         Sum += va_arg(Numbers, int);
11.      va_end(Numbers);
12.      return (Sum/Count);
13.  }
14.  int main()
15.  {
16.      float Average = avg(10, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9);
17.      cout << "Average of first 10 whole numbers : " << Average;
18.      return 0;
19.  }
```

2. How can you access the arguments that are manipulated in the function?

- a) `va_list`
- b) `arg_list`
- c) both `va_list` & `arg_list`
- d) none of the mentioned

View Answer

Answer: a

Explanation: `va_list` is provided by C++ to access manipulated arguments in function.

3. What is the output of this program?

```
1.      #include <iostream>
2.      #include <stdarg.h>
3.      using namespace std;
4.      float avg( int Count, ... )
5.      {
6.          va_list Numbers;
7.          va_start(Numbers, Count);
8.          int Sum = 0;
9.          for (int i = 0; i < Count; ++i )
10.             Sum += va_arg(Numbers, int);
11.          va_end(Numbers);
12.          return (Sum/Count);
13.      }
```

```
14.     int main()
15.     {
16.         float Average = avg(10, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9);
17.         cout << "Average of first 10 whole numbers : " << Average;
18.         return 0;
19.     }
```

a) 4

b) 5

c) 6

d) 7

View Answer

Answer: a

Explanation: We are just calculating the average of these numbers using `cstdarg`.

Output:

```
$ g++ uka.cpp
$ a.out
Average of first 10 whole numbers 4
```

3. What is the output of this program?

```
1.     #include <iostream>
2.     #include <stdarg.h>
3.     using namespace std;
4.     float avg( int Count, ... )
5.     {
6.         va_list Numbers;
7.         va_start(Numbers, Count);
8.         int Sum = 0;
9.         for (int i = 0; i < Count; ++i )
10.            Sum += va_arg(Numbers, int);
11.         va_end(Numbers);
12.         return (Sum/Count);
13.     }
14.     int main()
15.     {
16.         float Average = avg(10, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9);
17.         cout << "Average of first 10 whole numbers : " << Average;
18.         return 0;
19.     }
```

4. What is the maximum number of arguments or parameters that can be present in one function call?

- a) 64
- b) 256
- c) 255
- d) 16

[View Answer](#)

Answer: b

Explanation: C++ allows maximum number of 256 arguments in a function call.

3. What is the output of this program?

```
1.    #include <iostream>
2.    #include <stdarg.h>
3.    using namespace std;
4.    float avg( int Count, ... )
5.    {
6.        va_list Numbers;
7.        va_start(Numbers, Count);
8.        int Sum = 0;
9.        for (int i = 0; i < Count; ++i )
10.           Sum += va_arg(Numbers, int);
11.        va_end(Numbers);
12.        return (Sum/Count);
13.    }
14.    int main()
15.    {
16.        float Average = avg(10, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9);
17.        cout << "Average of first 10 whole numbers : " << Average;
18.        return 0;
19.    }
```

- a) 32
- b) 23
- c) 48
- d) compile time error

[View Answer](#)

Answer: a

Explanation: We are adding these numbers by using for statement and stdarg.

Output:

```
$ g++ uka.cpp
$ a.out
The result is 32
```

3. What is the output of this program?

```
1.  #include <iostream>
2.  #include <stdarg.h>
3.  using namespace std;
4.  float avg( int Count, ... )
5.  {
6.      va_list Numbers;
7.      va_start(Numbers, Count);
8.      int Sum = 0;
9.      for (int i = 0; i < Count; ++i )
10.         Sum += va_arg(Numbers, int);
11.      va_end(Numbers);
12.      return (Sum/Count);
13. }
14. int main()
15. {
16.     float Average = avg(10, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9);
17.     cout << "Average of first 10 whole numbers : " << Average;
18.     return 0;
19. }
```

- a) 2436
- b) 48697
- c) 1111111
- d) compile time error

View Answer

Answer: b

Explanation: In this program, we are eradicating the first value by comparing using while operator.

Output:

```
$ g++ rka3.cpp
$ a.out
48697
```

3. What is the output of this program?

```
1.  #include <iostream>
2.  #include <stdarg.h>
3.  using namespace std;
4.  float avg( int Count, ... )
5.  {
```

```
6.      va_list Numbers;
7.      va_start(Numbers, Count);
8.      int Sum = 0;
9.      for (int i = 0; i < Count; ++i )
10.         Sum += va_arg(Numbers, int);
11.      va_end(Numbers);
12.      return (Sum/Count);
13.  }
14.  int main()
15.  {
16.      float Average = avg(10, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9);
17.      cout << "Average of first 10 whole numbers : " << Average;
18.      return 0;
19.  }
```

- a) 6549
b) 4965
c) 6646
d) compile time error
View Answer

Answer: a

Explanation: In this program, we are returning the ascii value of the character and printing it.

Output:

```
$ g++ rka4.cpp
$ a.out
6549
```

3. What is the output of this program?

```
1.      #include <iostream>
2.      #include <stdarg.h>
3.      using namespace std;
4.      float avg( int Count, ... )
5.      {
6.          va_list Numbers;
7.          va_start(Numbers, Count);
8.          int Sum = 0;
9.          for (int i = 0; i < Count; ++i )
10.             Sum += va_arg(Numbers, int);
11.          va_end(Numbers);
```

```

12.         return (Sum/Count);
13.     }
14.     int main()
15.     {
16.         float Average = avg(10, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9);
17.         cout << "Average of first 10 whole numbers : " << Average;
18.         return 0;
19.     }

```

- a) stdlib.h
 - b) stdarg.h
 - c) string.h
 - d) stdpar.h
- View Answer

Answer: b

Explanation: <stdarg.h> header provided to perform variable number of argument passing.

3. What is the output of this program?

```

1.     #include <iostream>
2.     #include <stdarg.h>
3.     using namespace std;
4.     float avg( int Count, ... )
5.     {
6.         va_list Numbers;
7.         va_start(Numbers, Count);
8.         int Sum = 0;
9.         for (int i = 0; i < Count; ++i )
10.            Sum += va_arg(Numbers, int);
11.         va_end(Numbers);
12.         return (Sum/Count);
13.     }
14.     int main()
15.     {
16.         float Average = avg(10, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9);
17.         cout << "Average of first 10 whole numbers : " << Average;
18.         return 0;
19.     }

```

- a) 6

- b) 5
- c) 8
- d) 4

View Answer

Answer: d

Explanation: In this program, we are moving the pointer to the second value and printing it.

Output:

```
$ g++ uka6.cpp
$ a.out
4
```

3. What is the output of this program?

```
1.    #include <iostream>
2.    #include <stdarg.h>
3.    using namespace std;
4.    float avg( int Count, ... )
5.    {
6.        va_list Numbers;
7.        va_start(Numbers, Count);
8.        int Sum = 0;
9.        for (int i = 0; i < Count; ++i )
10.           Sum += va_arg(Numbers, int);
11.        va_end(Numbers);
12.        return (Sum/Count);
13.    }
14.    int main()
15.    {
16.        float Average = avg(10, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9);
17.        cout << "Average of first 10 whole numbers : " << Average;
18.        return 0;
19.    }
```

10. What will initialize the list of arguments in stdarg.h header file?

- a) va_list
- b) va_start
- c) va_arg
- d) none of the mentioned

View Answer

Answer: b

Explanation: va_start initialises the the list of arguments in <stdarg.h> header file.

C++ Programming Questions and Answers – Pointer to Function

1. To which does the function pointer point to?

- a) variable
- b) constants
- c) function
- d) absolute variables

View Answer

Answer: c

Explanation: A function pointer points to a function.

2. What will we not do with function pointers?

- a) allocation of memory
- b) deallocation of memory
- c) both allocation & deallocation of memory
- d) none of the mentioned

View Answer

Answer: c

Explanation: As it is used to execute a block of code, So we will not allocate or deallocate memory.

3. What is the default calling convention for a compiler in c++?

- a) __cdecl
- b) __stdcall
- c) __pascal
- d) __fastcall

View Answer

Answer: a

Explanation: __cdecl is the default calling convention for a compiler in c++.

a) 25

b) 35

c) 40

d) 45

View Answer

Answer: c

Explanation: In this program, we are adding two numbers with 15, So we got the output as 40.

Output:

```
$ g++ pfu2.cpp
$ a.out
40
```

a) 2

b) 20

c) 21

d) 22

View Answer

Answer: d

Explanation: As we are calling the function two times with the same value, So it is printing as 22.

Output:

```
$ g++ pfu.cpp
$ a.out
22
```

- a) d9
 - 9
 - b) d9d9
 - c) d9
 - d) compile time error
- [View Answer](#)

Answer: a

Explanation: As function pointer p is pointing to n(char, int), so for first call d9 will be printed for second call 10, which corresponds to '\n' character, and then 9 is printed.

Output:

```
$ g++ pfu1.cpp
$ a.out
d9
9
```

- a) 2323
 - b) 232
 - c) 23
 - d) compile time error
- [View Answer](#)

Answer: d

Explanation: In this program, we can't do the casting from char to int, So it is raising an error.

8. What is the mandatory part to present in function pointers?

- a) &
- b) return values
- c) data types
- d) none of the mentioned

[View Answer](#)

Answer: c

Explanation: The data types are mandatory for declaring the variables in the function pointers.

9. which of the following can be passed in function pointers?

- a) variables
- b) data types
- c) functions
- d) none of the mentioned

[View Answer](#)

Answer: c

Explanation: Only functions are passed in function pointers.

- a) ptr is pointer to function
- b) ptr is array of pointer to function
- c) ptr is pointer to such function which return type is array
- d) ptr is pointer to array of function

[View Answer](#)

Answer: b

Explanation: In this expression, ptr is array not pointer.

ayzom.com

C++ Programming Questions and Answers – Macros

1. which keyword is used to define the macros in c++?

- a) macro
- b) define
- c) #define
- d) none of the mentioned

View Answer

Answer: c

Explanation: #define is the keyword which is used to define the macros in c++.

2. Which symbol is used to declare the preprocessor directives?

- a) #
- b) \$
- c) *
- d) ^

View Answer

Answer: a

Explanation: # symbol is used to declare the preprocessor directives.

3. How many types of macros are there in c++?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: b

Explanation: There are two types of macros. They are object-like and function-like.

4. What is the mandatory preprocessor directive for c++?

- a) #define <iostream>
- b) #include <iostream>
- c) #undef <iostream>
- d) none of the mentioned

View Answer

Answer: b

Explanation: For a c++ program to execute, we need #include<iostream>.

- a) 100.01
- b) 100.1
- c) compile time error
- d) none of the mentioned

View Answer

Answer: a

Explanation: In this program, we are getting the minimum number using conditional operator.

Output:

```
$ g++ mac3.cpp
$ a.out
The minimum value is 100.01
```

- a) 5
- b) details about your file
- c) compile time error
- d) none of the mentioned

View Answer

Answer: b

Explanation: In this program, we are using the macros to print the information about the file.

Output:

```
$ g++ mac2.cpp
$ a.out
Value of __LINE__ : 5
Value of __FILE__ : mac1.cpp
Value of __DATE__ : Oct 10 2012
Value of __TIME__ : 22:24:37
```

- a) 16
- b) 64
- c) compile time error
- d) none of the mentioned

View Answer

Answer: d

Explanation: In this program, as we have not initialize the variable x, we will get a output of ending digit of 4.

Output:

```
$ g++ mac1.cpp
$ a.out
75386824
```

- a) 10
- b) 15
- c) 20
- d) none of the mentioned

View Answer

Answer: a

Explanation: In this program, we are just printing the declared values.

Output:

```
$ g++ mac.cpp
$ a.out
10
```

- a) 11
- b) 10
- c) compile time error
- d) none of the mentioned

View Answer

Answer: c

Explanation: Macro Preprocessor only replaces occurrence of macro symbol with macro symbol value, So we can't increment the value.

10. What is the other name of the macro?

- a) scripted directive
- b) executed directive
- c) link directive
- d) none of the mentioned

View Answer

Answer: a

Explanation: When the compiler encounters a previously defined macro, it will take the result from that execution itself.

ayzom.com

C++ Programming Questions and Answers – Modularization and Interfaces

1. which of the following is used to implement the c++ interfaces?

- a) absolute variables
- b) abstract classes
- c) constant variables
- d) none of the mentioned

View Answer

Answer: b

Explanation: Abstract classes in C++ are purposely defined for making base classes containing atleast one virtual function which can be overloaded on inheritance, which means single function name for different sub-classes, hence acts as an interface.

2. What is the ability to group some lines of code that can be included in the program?

- a) specific task
- b) program control
- c) modularization
- d) macros

View Answer

Answer: c

Explanation: Modularization is also similar to macros but it is used to build large projects.

3. How many types do functions fall depends on modularization?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: b

Explanation: There are two types of functions. They are program control and specific task.

4. How many types of modularization are there in c++?

- a) 4
- b) 3
- c) 1
- d) none of the mentioned

View Answer

Answer: d

Explanation: There are two types of modular programming. They are interface and implementation.

5. What does the client module import?

- a) macro
- b) records
- c) interface
- d) none of the mentioned

View Answer

Answer: c

Explanation: Because they access the functions in the module user interface.

6. Identify the correct statement.

- a) c++ does not have built-in interfaces
- b) c++ does have built-in interfaces
- c) c++ have no concept of interfaces
- d) none of the mentioned

View Answer

Answer: a

Explanation: Unlike other programming languages like Java and others, C++ has no inbuilt interfaces.

7. What is similar to the interface in c++

- a) methods
- b) instance of a class
- c) pure abstract class
- d) none of the mentioned

View Answer

Answer: c

Explanation: Pure abstract classes in C++ are a type of interface because it contains only abstract member functions and no data or concrete member functions.

8. Which of the following implements the module in the program?

- a) macro
- b) header files
- c) macro & header files
- d) none of the mentioned

View Answer

Answer: b

Explanation: We can include the group of code by using the #include header file.

C++ Programming Questions and Answers – Namespaces

1. Which operator is used to signify the namespace?

- a) conditional operator
- b) ternary operator
- c) scope operator
- d) none of the mentioned

View Answer

Answer: c

Explanation: Scope operator(::) is used in namespace syntax.

General syntax:

```
namespace X{ int a;}  
cout<<X::a;
```

2. Identify the correct statement.

- a) Namespace is used to group class, objects and functions
- b) Namespace is used to mark the beginning of the program
- c) A namespace is used to separate the class, objects
- d) None of the mentioned

View Answer

Answer: a

Explanation: Namespace allows you to group class, objects, and functions. It is used to divide the global scope into the sub-scopes.

3. What is the use of Namespace?

- a) To encapsulate the data
- b) To structure a program into logical units
- c) Encapsulate the data & structure a program into logical units
- d) None of the mentioned

View Answer

Answer: b

Explanation: The main aim of the namespace is to understand the logical units of the program and to make the program so robust.

4. What is the general syntax for accessing the namespace variable?

- a) namespace::operator
- b) namespace,operator
- c) namespace#operator
- d) none of the mentioned

View Answer

Answer: a

Explanation: To access variables from namespace we use following syntax.

namespace :: variable;

General syntax:

```
namespace X{ int a;}  
cout<<X::a;
```

a) 8.31416

b) 8

c) 9

d) compile time error

View Answer

Answer: b

Explanation: As we are getting two variables from namespace variable and we are adding that.

Output:

```
$ g++ name.cpp
$ a.out
8
```

a) 11

b) 01

c) 00

d) 10

View Answer

Answer: d

Explanation: We are inter mixing the variable and comparing it which is bigger and smaller and according to that we are printing the output.

Output:

```
$ g++ name1.cpp
$ a.out
10
```

a) 4

b) 13

c) 16

d) compile time error

View Answer

Answer: c

Explanation: In this program, as there is lot of variable a and it is printing the value inside the block because it got the highest priority.

Output:

```
$ g++ name2.cpp
$ a.out
16
```

a) 1015

b) 1510

c) 55

d) compile time error

View Answer

Answer: c

Explanation: We are overriding the value at the main function and so we are getting the output as 55.

Output:

```
$ g++ name4.cpp
$ a.out
55
```

a) 9

b) 10

c) compile time error

d) none of the mentioned

View Answer

Answer: a

Explanation: A scope resolution operator without a scope qualifier refers to the global namespace.

10. Which keyword is used to access the variable in the namespace?

- a) using
- b) dynamic
- c) const
- d) static

View Answer

Answer: a

Explanation: using keyword is used to specify the name of the namespace to which the variable belongs.

eg.

```
namespace A { int a = 5; }
```

```
namespace B { int a = 10; }
```

```
using namespace A;
```

```
cout<<a; // will print value of a from namespace A.
```

```
using namespace B;
```

```
cout<<a; // will print value of a from namespace B.
```

ayzom.com

C++ Programming Questions and Answers – Namespaces – 2

1. Pick the incorrect statement for namespaces in C++.

- a) Namespace declarations are always global scope
- b) Keyword namespace is used at the starting of a namespace definition
- c) Namespace has access specifiers like private or public
- d) Namespace definitions can be nested

View Answer

Answer: c

Explanation: Namespace does not have any specifiers associated with it like classes or structures.

2. Which operator is used for accessing a member of namespace?

- a) :
- b) ::
- c) ->
- d) .

View Answer

Answer: b

Explanation: Scope resolution operator(::) is used for accessing a member of a namespace. example:

```
namespace A{  
    int var;  
}  
A::var = 5;
```

View Answer

Answer: c

Explanation: A namespace definition always starts with the namespace keyword so definition with Namespace(capital N) is wrong. namespace does not terminate by a semicolon hence the definition with a semicolon is wrong. every variable declaration in C++ should end with semicolon therefore namespace containing 'int i' without semicolon is wrong.

- a) 10
- b) 5
- c) Error
- d) 105

View Answer

Answer: c

Explanation: Variable cout is defined in above defined namespace B and also in the inbuilt namespace std. So the compiler confuses and throws an error saying that cout is ambiguous i.e. which cout to use as it is available in both std and B namespace.

- a) 5
- b) 10
- c) Error
- d) Wrong use of namespace

View Answer

Answer: a

Explanation: As we have mentioned that 'using namespace B' so now wherever var will be used it will be from namespace B. hence the output was the value of var from namespace B.

- a) 5
- b) 10
- c) 20

d) Error

View Answer

Answer: c

Explanation: As var is already declared in this scope so that gets preference over others. Therefore 20 is printed which is the value assigned to var declared in this scope.

a) 10

b) Error

c) Some garbage value

d) Nothing but program runs perfectly

View Answer

Answer: a

Explanation: A namespace without name is called unnamed namespace and is valid in that scope only. So its like global scope of variable. One can access that var from main() function.

8. What is the correct syntax of defining a namespace?

a) namespace name {}

b) Namespace name {};

c) namespace name {};

d) typedef namespace name {} NAME

View Answer

Answer: a

Explanation: A namespace:

- Starts with keyword namespace

- Followed by identifier

- All members inside the braces {}

- No semicolon at the end

namespace identifier {} .

a) cout<<A::i;

b) cout<<B::i;

c) cout<<A::B::i;

d) cout<<i;

View Answer

Answer: c

Explanation: Here namespace B is nested inside the namespace A. Hence to access the variable i we need to mention through B and A. So it should A::B::i, which means i belongs to namespace B which is defined inside the namespace A.

a) 9999999999

b) 1111111111

c) error

d) segmentation fault

View Answer

Answer: a

Explanation: C++ allows to use namespaces aliases i.e. if a namespace having a large name we can assign it to a new namespace having a small name for the convenience in coding. This assignment of namespaces is called namespace aliasing.

c) Error due to clash of func()

d) This is not allowed in C++

View Answer

Answer: b

Explanation: Here both the func() are available in different namespaces having same name but different types which is equivalent to function overloading. So no error will be there as function overloading is allowed in C++.

View Answer

Answer: b

Explanation: Here we have specified that func() should be called from the namespace A, hence both the calls will use the same function from namespace A.

What changes you can do in the header files to avoid the redefinition that compiler will give when both the header files are included in the same program keeping the declaration of both the functions same?

- a) Cannot be handled because C++ does not allow this
- b) Declare both the function inside different namespaces
- c) Include one header files where they are needed so that no clashes occur
- d) Make the header files name same

View Answer

Answer: b

Explanation: Define both the function in different namespaces under their respective header files. So whenever we need them we can use the name of their respective namespaces to call them. This will resolve the error keeping function declarations same.

Modified Header files :

```
h1.h
Content of h1.h
-----
#include <iostream>
using namespace std;
namespace A{
    int func(int a){
        cout<<"Multiplied by 2";
        return 2*a;
    }
}
-----
```

```
Content of h2.h
-----
#include <iostream>
using namespace std;
namespace B{
    float func(float a){
        cout<<"divided by 2";
        return a/2;
    }
}
-----
```

Now one can use multiplication func as A::func(int); and division function as B::func(int).

C++ Programming Questions and Answers – Exceptions

1. To where does the program control transfers when the exception is arisen?

- a) catch
- b) handlers
- c) throw
- d) none of the mentioned

View Answer

Answer: b

Explanation: When an exception is arisen mean, the exception is caught by handlers and then it decides the type of exception.

2. Which keyword is used to check exception in the block of code?

- a) catch
- b) throw
- c) try
- d) none of the mentioned

View Answer

Answer: c

Explanation: The try() statement is used for exceptions in c++.

3. What will happen when the exception is not caught in the program?

- a) error
- b) program will execute
- c) block of that code will not execute
- d) none of the mentioned

View Answer

Answer: a

Explanation: When exceptions are not caught in any program then program throws error.

- a) 0
- b) error:Positive Number Required
- c) compile time error
- d) none of the mentioned

View Answer

Answer: a

Explanation: As the zero marks the beginning of the positive number, it is printed as output

Output:

```
$ g++ excep.cpp
$ a.out
0
```

- a) compile time error
- b) prints first 19 numbers
- c) prints first 19 numbers and throws exception at 20
- d) none of the mentioned

View Answer

Answer: c

Explanation: In this program, we are printing upto 19 numbers and when executing the 20, we are raising a exception.

Output:


```
$ g++ excep1.cpp
$ a.out
12345678910111213141516171819Caught an exception with value: 20
```

- a) 25
- b) 20
- c) Division by zero condition!
- d) None of the mentioned

View Answer

Answer: a

Explanation: In this program, we resembling the division by using the exception handling.

Output:

```
$ g++ excep2.cpp
$ a.out
25
```

- a) 4 Bytes allocated successfully
- b) 8 Bytes allocated successfully
- c) Memory allocation failure
- d) Depends on the size of the data type

View Answer

Answer: d

Explanation: As we are allocating the memory to the variables and if there are not sufficient size means, it will throw an exception.

Output:

```
$ g++ excep3.cpp
$ a.out
4 Bytes allocated successfully
```

- a) caught a double type
- b) compile time error
- c) abnormal program termination
- d) none of the mentioned

View Answer

Answer: c

Explanation: As we are throwing integer to double it will raise as abnormal program after termination throw statement.

Output:

```
$ g++ excep4.cpp
$ a.out
terminate called after throwing an instance of 'int'
Aborted
```

- a) Allocated successfully
- b) Error allocating the requested memory
- c) Depends on the memory of the computer
- d) None of the mentioned

View Answer

Answer: c

Explanation: In this program, we allocating the memory to the arrays by using exception handling and we handled the exception by standard exception.

Output:

```
$ g++ excep5.cpp  
$ a.out  
Allocated successfully
```

10. What will happen when the handler is not found for an exception?

- a) calls the standard library function terminate()
- b) raise an error
- c) executes the remaining block
- d) none of the mentioned

View Answer

Answer: a

Explanation: None.

ayzom.com

C++ Programming Questions and Answers – Linkage

1. How many types of linkages are there in C++?

- a) 1
- b) 2
- c) 3
- d) 4

[View Answer](#)

Answer: c

Explanation: There are three types of linkage in c++. They are an internal linkage, external linkage, and no linkage.

2. To use internal linkage we have to use which keyword?

- a) static
- b) extern
- c) static or extern
- d) none of the mentioned

[View Answer](#)

Answer: a

Explanation: None.

3. Which one is used to refer to program elements in any translation units?

- a) internal linkage
- b) external linkage
- c) no linkage
- d) none of the mentioned

[View Answer](#)

Answer: b

Explanation: In the external linkage, it is used to refer to identifiers in various programs.

- a) 842
- b) 843
- c) compile time error
- d) none of the mentioned

[View Answer](#)

Answer: a

Explanation: In this program, we have created a header file and linked that into the source program and we post incrementing that because of that it is printed as 842.

Output:

```
$ g++ link.cpp
$ a.out
842
```

5. What is the default type of linkage that is available for identifiers?

- a) internal
- b) external
- c) no linkage
- d) none of the mentioned

[View Answer](#)

Answer: b

Explanation: None.

6. To use external linkage we have to use which keyword?

- a) static
- b) extern
- c) const
- d) none of the mentioned

View Answer

Answer: b

Explanation: Extern keyword is used to represent identifiers from other programs.

7. Which is used to use a function from one source file to another?

- a) code
- b) declaration
- c) prototype
- d) none of the mentioned

View Answer

Answer: c

Explanation: By defining a function's prototype in another file means, we can inherit all the features from the source function.

8. What is the use of no linkage?

- a) make the entity visible to other programs
- b) make the entity visible to other blocks in the same program.
- c) make the entity visible only to that block
- d) none of the mentioned

View Answer

Answer: c

Explanation: None.

C++ Programming Questions and Answers – Header Files Usage

1. What is the user-defined header file extension in c++?

- a) cpp
- b) h
- c) hf
- d) none of the mentioned

View Answer

Answer: b

Explanation: .h extensions are used for user defined header files. To include a user defined header file one should use `#include "name.h"` i.e. enclosed within double quotes.

2. Which of the following keyword is used to declare the header file?

- a) include
- b) exclude
- c) string
- d) namespace

View Answer

Answer: a

Explanation: The include keyword is used to include all the required things to execute the given code in the program.

3. Identify the incorrect statement.

- a) iostream is a standard header and iostream.h is a non-standard header
- b) iostream is a non-standard header and iostream.h is a non-standard header
- c) iostream is a standard header and iostream.h is a standard header
- d) none of the mentioned

View Answer

Answer: a

Explanation: The iostream.h is used in the older versions of c++ and iostream is evolved from it in the std namespace.

4. What does a default header file contain?

- a) prototype
- b) implementation
- c) declarations
- d) none of the mentioned

View Answer

Answer: c

Explanation: In the header file, we define something that to be manipulated in the program.

- a) jobsjobs
- b) jobs
- c) compile time error
- d) program will not run

View Answer

Answer: c

Explanation: This program will run on older version of C++ with the inclusion of `#include` header file, but for on new compiler C++14 and above the gets is removed from the header file so it will not run on them even after inclusion of `cstdio` header file.

6. setprecision requires which of the following header file?

- a) stdlib.h

b) iomanip.h
c) console.h
d) conio.h
View Answer

Answer: b

Explanation: The iomanip header file is used to correct the precision of the values.

7. Which of the following header file does not exist?

a) <iostream>
b) <string>
c) <sstring>
d) <sstream>
View Answer

Answer: c

Explanation: There is no such header file <sstring> in C++.

8. Which of the header file must be included to use stringstream?

a) <iostream>
b) <string>
c) <sstring>
d) <sstream>
View Answer

Answer: b

Explanation: stringstream is available under the header file <string> in C++.

9. Which of the following header files is required for creating and reading data files?

a) ofstream.h
b) fstream.h
c) ifstream.h
d) console.h
View Answer

Answer: b

Explanation: In this fstream.h header file is used for accessing the files only.

a) 4
b) 5
c) 6
d) compile time error
View Answer

Answer: a

Explanation: In this program, we are finding the average of first 10 numbers using stdarg header file

Output:

```
$ g++ std.cpp  
$ a.out  
4
```

C++ Programming Questions and Answers – Classes

1. What does a class in C++ holds?

- a) data
- b) functions
- c) both data & functions
- d) none of the mentioned

View Answer

Answer: c

Explanation: The classes in C++ encapsulates(i.e. put together) all the data and functions related to them for manipulation.

2. How many specifiers are present in access specifiers in class?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: c

Explanation: There are three types of access specifiers. They are public, protected and private.

3. Which is used to define the member of a class externally?

- a) :
- b) ::
- c) #
- d) none of the mentioned

View Answer

Answer: b

Explanation: :: operator is used to define the body of any class function outside the class.

4. Which other keywords are also used to declare the class other than class?

- a) struct
- b) union
- c) object
- d) both struct & union

View Answer

Answer: d

Explanation: Struct and union take the same definition of class but differs in the access techniques.

- a) rect area: 24
- b) rect area: 12
- c) compile error because rect is as used as class name and variable name in line #20
- d) none of the mentioned

View Answer

Answer: b

Explanation: In this program, we are calculating the area of rectangle based on given values.

Output:

```
$ g++ class.cpp
$ a.out
rect area: 12
```

- a) execute
- b) not execute
- c) none of the mentioned
- d) both execute & not execute

View Answer

Answer: a

Explanation: In this program, we are just pointing the pointer to a object and printing execute if it is correctly pointed.

Output:

```
$ g++ class1.cpp
$ a.out
execute
```

7. Which of the following is a valid class declaration?

- a) class A { int x; };
- b) class B { }
- c) public class A { }
- d) object A { int x; };

View Answer

Answer: a

Explanation: A class declaration terminates with semicolon and starts with class keyword. only option (a) follows these rules therefore class A { int x; }; is correct.

8. The data members and functions of a class in C++ are by default _____

- a) protected
- b) private
- c) public
- d) none of the mentioned

View Answer

Answer: b

Explanation: By default all the data members and member functions of class are private.

9. Constructors are used to

- a) initialize the objects
- b) construct the data members
- c) both initialize the objects & construct the data members
- d) none of the mentioned

View Answer

Answer: a

Explanation: Once the object is declared means, the constructor are also declared by default.

10. When struct is used instead of the keyword class means, what will happen in the program?

- a) access is public by default
- b) access is private by default
- c) access is protected by default
- d) none of the mentioned

View Answer

Answer: a

Explanation: For structures, by default all the data members and member functions are public.

C++ Programming Questions and Answers – Classes – 2

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
class A
{
    int a;

public:
    int assign(int i) const {
        a = i;
    }
    int return_value() const {
        return a;
    }
};
int main(int argc, char const *argv[])
{
    A obj;
    obj.assign(5);
    cout<<obj.return_value();
}
```

1. Which category of data type a class belongs to?

- a) Fundamental data type
- b) Derived data type
- c) User defined derived data type
- d) Atomic data type

View Answer

Answer: c

Explanation: Fundamental/Atomic data type includes int, char, float, double and void. Derived data type includes arrays, pointers, references, function and constants. User defined derived data type includes class, structure, union and enumeration.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
class A
{
    int a;

public:
    int assign(int i) const {
        a = i;
    }
    int return_value() const {
        return a;
    }
};
int main(int argc, char const *argv[])
{
    A obj;
    obj.assign(5);
    cout<<obj.return_value();
}
```

2. Which operator a pointer object of a class uses to access its data members and member functions?

- a) .
- b) ->
- c) :

d) ::

[View Answer](#)

Answer: b

Explanation: ->(arrow operator) is used by a pointer object to access members of its class.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
class A
{
    int a;

public:
    int assign(int i) const {
        a = i;
    }
    int return_value() const {
        return a;
    }
};
int main(int argc, char const *argv[])
{
    A obj;
    obj.assign(5);
    cout<<obj.return_value();
}
```

a) 5

b) 10

c) Error

d) Segmentation fault

[View Answer](#)

Answer: c

Explanation: As the assign() is a constant function and a constant function cannot change the state of an object and as in the assign function we are trying to modify the member a of the object therefore the program gives error.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
class A
{
    int a;

public:
    int assign(int i) const {
        a = i;
    }
    int return_value() const {
        return a;
    }
};
int main(int argc, char const *argv[])
{
    A obj;
    obj.assign(5);
    cout<<obj.return_value();
}
```

a) A.value

b) A::value

- c) A->value
 d) A^value
 View Answer

Answer: b

Explanation: Scope resolution operator(::) is used to access a static member of a class.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
class A
{
    int a;

public:
    int assign(int i) const {
        a = i;
    }
    int return_value() const {
        return a;
    }
};
int main(int argc, char const *argv[])
{
    A obj;
    obj.assign(5);
    cout<<obj.return_value();
}
```

5. How the objects are self-referenced in a member function of that class.

- a) Using a special keyword object
 b) Using this pointer
 c) Using * with the name of that object
 d) By passing self as a parameter in the member function

View Answer

Answer: b

Explanation: In Classes objects are self-referenced using this pointer inside the member functions. for example this->value to access the data member value of that object.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
class A
{
    int a;

public:
    int assign(int i) const {
        a = i;
    }
    int return_value() const {
        return a;
    }
};
int main(int argc, char const *argv[])
{
    A obj;
    obj.assign(5);
    cout<<obj.return_value();
}
```

6. What does a mutable member of a class mean?

- a) A member that can never be changed
- b) A member that can be updated only if it not a member of constant object
- c) A member that can be updated even if it a member of constant object
- d) A member that is global throughout the class

View Answer

Answer: c

Explanation: Mutable members are those which can be updated even if it a member of a constant object. You can change their value even from a constant member function of that class.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
class A
{
    int a;

public:
    int assign(int i) const {
        a = i;
    }
    int return_value() const {
        return a;
    }
};
int main(int argc, char const *argv[])
{
    A obj;
    obj.assign(5);
    cout<<obj.return_value();
}
```

- a) 5
- b) Error
- c) Segmentation fault
- d) Undefined value

View Answer

Answer: a

Explanation: As a is a mutable member of the class its value can be modified whether it is a part of constant object or not. It can be modified even inside a constant member function. Hence, the program runs fine and does not give any error.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
class A
{
    int a;

public:
    int assign(int i) const {
        a = i;
    }
    int return_value() const {
        return a;
    }
};
int main(int argc, char const *argv[])
{
    A obj;
```

```

    obj.assign(5);
    cout<<obj.return_value();
}

```

8. Pick the incorrect statement about inline functions in C++?

- a) They reduce function call overheads
- b) These functions are inserted/substituted at the point of call
- c) Saves overhead of a return call from a function
- d) They are generally very large and complicated function

View Answer

Answer: d

Explanation: Inline are functions that are expanded when it is called. The whole code of the inline function gets inserted/substituted at the point of call. In this, they help in reducing the function call overheads. Also they save overhead of a return call from a function. Inline functions are generally kept small.

3. What is the output of the following C++ code?

```

#include <iostream>
#include <string>
using namespace std;
class A
{
    int a;

public:
    int assign(int i) const {
        a = i;
    }
    int return_value() const {
        return a;
    }
};
int main(int argc, char const *argv[])
{
    A obj;
    obj.assign(5);
    cout<<obj.return_value();
}

```

9. Inline functions are avoided when _____

- a) function contains static variables
- b) function have recursive calls
- c) function have loops
- d) all of the mentioned

View Answer

Answer: d

Explanation: Inline functions are avoided in all the above cases as whole inline code is copied to the point of call so compiler avoids to make large functions as inline. Even if you yourself mention inline but the function is large compiler ignores your request of inline and treats that function as a normal function.

3. What is the output of the following C++ code?

```

#include <iostream>
#include <string>
using namespace std;
class A
{
    int a;

public:
    int assign(int i) const {
        a = i;
    }
}

```

```

        int return_value() const {
            return a;
        }
};
int main(int argc, char const *argv[])
{
    A obj;
    obj.assign(5);
    cout<<obj.return_value();
}

```

10. Pick the correct statement.

- a) Macros and inline functions are same thing
- b) Macros looks like function calls but they are actually not
- c) Inline functions looks like function but they are not
- d) Inline function are always large

View Answer

Answer: b

Explanation: Macros in C++ looks like function calls but actually they are not function calls.

3. What is the output of the following C++ code?

```

#include <iostream>
#include <string>
using namespace std;
class A
{
    int a;

public:
    int assign(int i) const {
        a = i;
    }
    int return_value() const {
        return a;
    }
};
int main(int argc, char const *argv[])
{
    A obj;
    obj.assign(5);
    cout<<obj.return_value();
}

```

- a) Hello World
- b) Error
- c) Segmentation Fault
- d) Blank Space

View Answer

Answer: b

Explanation: Macros cannot access the private member of a class therefore #define MAC(S:m) will give an error.

3. What is the output of the following C++ code?

```

#include <iostream>
#include <string>
using namespace std;
class A
{
    int a;

public:
    int assign(int i) const {
        a = i;
    }
}

```

```

    }
    int return_value() const {
        return a;
    }
};
int main(int argc, char const *argv[])
{
    A obj;
    obj.assign(5);
    cout<<obj.return_value();
}

```

- a) 5
- b) Garbage value
- c) Error
- d) Segmentation fault

View Answer

Answer: c

Explanation: Every static member of a class is initialised before its use. As 'a' is a static member of the class and is not initialised so the program will give error.

3. What is the output of the following C++ code?

```

#include <iostream>
#include <string>
using namespace std;
class A
{
    int a;

public:
    int assign(int i) const {
        a = i;
    }
    int return_value() const {
        return a;
    }
};
int main(int argc, char const *argv[])
{
    A obj;
    obj.assign(5);
    cout<<obj.return_value();
}

```

- a) 1055
- b) 555
- c) 101010
- d) 51010

View Answer

Answer: c

Explanation: As 'a' is a static member of the class so it is a type of global variable to the class i.e. any change made by one object is reflected back to all the other objects. Hence when a is changed to 10 by object a1, so value of 'a' becomes 10 for each object and 3 times 10 is printed.

3. What is the output of the following C++ code?

```

#include <iostream>
#include <string>
using namespace std;
class A
{
    int a;

```

```

public:
    int assign(int i) const {
        a = i;
    }
    int return_value() const {
        return a;
    }
};
int main(int argc, char const *argv[])
{
    A obj;
    obj.assign(5);
    cout<<obj.return_value();
}

```

- a) 10
- b) Error
- c) Segmentation Fault
- d) 5

View Answer

Answer: b

Explanation: As value_of_a() is a static function and static member can access only static members therefore the program will give error.

3. What is the output of the following C++ code?

```

#include <iostream>
#include <string>
using namespace std;
class A
{
    int a;

public:
    int assign(int i) const {
        a = i;
    }
    int return_value() const {
        return a;
    }
};
int main(int argc, char const *argv[])
{
    A obj;
    obj.assign(5);
    cout<<obj.return_value();
}

```

15. Which functions of a class are called inline functions?

- a) All the functions containing declared inside the class
- b) All functions defined inside or with the inline keyword
- c) All the functions accessing static members of the class
- d) All the functions that are defined outside the class

View Answer

Answer: b

Explanation: All the functions defined inside the class or functions having inline keyword before them are inline functions of a class provided they are small and simple otherwise compiler ignores the request of inline.

C++ Programming Questions and Answers – User Defined Types

1. Which keyword is used to define the user defined data types?

- a) def
- b) union
- c) typedef
- d) type

View Answer

Answer: c

Explanation: Typedef is used to define user defined datatypes.

eg:

typedef int INT;

INT a;

here INT is used defined data type.

2. Identify the correct statement.

- a) typedef does not create different types. It only creates synonyms of existing types
- b) typedef create different types
- c) typedef create own types
- d) none of the mentioned

View Answer

Answer: a

Explanation: By using typedef, we can create a type of pre-existing type only not our own type of data.

3. What does the data type defined by union will do?

- a) It allow one different portion of memory to be accessed as same data types
- b) It allow one same portion of memory to be accessed as same data types
- c) It allow one different portion of memory to be accessed as different data types
- d) It allow one same portion of memory to be accessed as different data types

View Answer

Answer: d

Explanation: Union is used to define the data types of our choice and it will store the data type in one location make them accessible.

a) 20

b) 15

c) 30

d) 25

View Answer

Answer: a

Explanation: In this program, we are manipulating the numbers and printing the result using user-defined data types.

Output:

```
<pre lang="cpp" cssfile="hk1_style">
$ g++ user.cpp
$ a.out
20
```

a) 012345678910

b) 0123456789

c) 01234567891011

d) none of the mentioned

View Answer

Answer: a

Explanation: In this program, we are defined the data types as enumerator and printing its value in a order.

Output:

```
$ g++ user1.cpp
$ a.out
012345678910
```

- a) 10steve
 - b) steve10
 - c) compile time error
 - d) compile but not run
- View Answer

Answer: c

Explanation: Error: invalid conversion from 'const char*' to 'let {aka char}'.

7. What is the syntax of user-defined data types?

- a) typedef ExistingDataType NameByUser
 - b) typedef NameByUser ExistingDataType
 - c) def NameByUser ExistingDataType
 - d) none of the mentioned
- View Answer

Answer: a

Explanation: correct syntax is typedef ExistingDataType NameByUser.
typedef int INT; (typedef existing-datatype New-name;).

8. How many types of user-defined data type are in c++?

- a) 1
 - b) 2
 - c) 3
 - d) 4
- View Answer

Answer: c

Explanation: There are three types of user-defined data types. They are typedef, union, enumerator.

9. What is the scope of typedef defined data types?

- a) inside that block only
 - b) whole program
 - c) outside the program
 - d) none of the mentioned
- View Answer

Answer: b

Explanation: We are defining the user-defined data type to be availed only inside that program, if we want to use anywhere means we have to define those types in the header file.

10. How many types of models are available to create the user-defined data type?

- a) 1
 - b) 2
 - c) 3
 - d) 4
- View Answer

Answer: b

Explanation: There are two types of models. They are references to built-in types and multipart types.

ayzom.com

C++ Programming Questions and Answers – Objects

1. Where does the object is created?

- a) class
- b) constructor
- c) destructor
- d) attributes

View Answer

Answer: a

Explanation: In class, only all the listed items except class will be declared.

2. How to access the object in the class?

- a) scope resolution operator
- b) ternary operator
- c) direct member access operator
- d) none of the mentioned

View Answer

Answer: c

Explanation: Objects in the method can be accessed using direct member access operator which is (.).

3. Which of these following members are not accessed by using direct member access operator?

- a) public
- b) private
- c) protected
- d) both private & protected

View Answer

Answer: d

Explanation: Because of the access is given to the private and protected, We can't access them by using direct member access operator.

a) 210

b) 213

c) 215

d) 217

View Answer

Answer: b

Explanation: In the above program, we are calculating the area of the cube by using the cube formula

Output:

```
$ g++ obj1.cpp
$ a.out
213
```

a) recta area: 30 rectb area: 42

b) recta area: 20 rectb area: 34

c) recta area: 30 rectb area: 21

d) none of the mentioned

View Answer

Answer: a

Explanation: We are calculating the area of rectangle by two objects.

6. Pick out the other definition of objects.

- a) member of the class
- b) associate of the class
- c) attribute of the class
- d) instance of the class

View Answer

Answer: d

Explanation: An Object represents an instance of a class i.e. a variable of that class type having access to its data members and member functions from outside if allowed.

7. How many objects can present in a single class?

- a) 1
- b) 2
- c) 3
- d) as many as possible

View Answer

Answer: d

Explanation: Because a class may contain any number of objects according to its compliance.

a) Enter an integer 5

Variable entered is 5

- b) Runtime error
- c) Error
- d) None of the mentioned

View Answer

Answer: c

Explanation: There is no member function var() in the class hence the program will through an error stating var is a private data member and it cannot be used as a function.

9. Which special character is used to mark the end of class?

- a) ;
- b) :
- c) #
- d) \$

View Answer

Answer: a

Explanation: Similar to ending any statement, a class is also terminated with semicolon(;).

a) 10

b) 11

c) 20

d) 22

View Answer

Answer: a

Explanation: We are getting the number and copying it to j and printing it.

Output:

```
$ g++ obj2.cpp
$ a.out
10
```

C++ Programming Questions and Answers – Operator Functions

1. Pick the other name of operator function.

- a) function overloading
- b) operator overloading
- c) member overloading
- d) none of the mentioned

View Answer

Answer: b

Explanation: Operator function means operation defined for that operator so if user defines a function for an operator then that is called operator overloading i.e. overloading already present operator function.

2. Which of the following operators can't be overloaded?

- a) ::
- b) +
- c) -
- d) [].

View Answer

Answer: a

Explanation: :: operator cannot be overloaded because this operator operates on names rather than values and C++ has no syntax for writing codes that works on names than values so using syntax these operators cannot be overloaded.

3. How to declare operator function?

- a) operator sign
- b) operator
- c) name of the operator
- d) none of the mentioned

View Answer

Answer: a

Explanation: We have to declare the operator function by using the operator, operator sign. Example "operator +" where the operator is a keyword and + is the symbol need to be overloaded.

- a) 5, 5
- b) 7, 3
- c) 3, 7
- d) None of the mentioned

View Answer

Answer: b

Explanation: In this program, we are adding the first number of a with first number of b by using operator function and also we are adding second number by this method also.

Output:

```
$ g++ oper.cpp
$ a.out
7, 3
```

a) Volume of Box1 : 210

Volume of Box2 : 1560

Volume of Box3 : 5400

b) Volume of Box1 : 200

Volume of Box2 : 1560

Volume of Box3 : 5400

- c) Volume of Box1 : 210
Volume of Box2 : 1550
Volume of Box3 : 5400
d) None of the mentioned
View Answer

Answer: a

Explanation: In this program, we finding the box3 area by adding box1 and box2.

Output:

```
$ g++ oper1.cpp
$ a.out
Volume of Box1 : 210
Volume of Box2 : 1560
Volume of Box3 : 5400
```

- a) operator+
operator+=
b) operator+=
operator+
c) operator+
operator+
d) none of the mentioned
View Answer

Answer: a

Explanation: We are using two operator functions and executing them and the result is printed according to the order.

Output:

```
$ g++ oper2.cpp
$ a.out
operator+
operator+=
```

- a) 10 10
b) 11 11
c) error
d) runtime error
View Answer

Answer: a

Explanation: In this program, -> operator is used to describe the member of the class and so we are getting this output.

Output:

```
$ g++ char4.cpp
$ a.out
10 10
```

8. Which of the following statements is NOT valid about operator overloading?
a) Only existing operators can be overloaded
b) The overloaded operator must have at least one operand of its class type
c) The overloaded operators follow the syntax rules of the original operator
d) None of the mentioned
View Answer

Answer: d

Explanation: The overloaded operator must not have at least one operand of its class type.

9. Operator overloading is

- a) making c++ operator works with objects
- b) giving new meaning to existing operator
- c) making the new operator
- d) adding operation to the existing operators

View Answer

Answer: d

Explanation: Operator overloading is the way adding operation to the existing operators.

- a) 5
- b) 6
- c) error
- d) runtime error

View Answer

Answer: c

Explanation: In this program, there will arise an ambiguous overload for 5.

ayzom.com

C++ Programming Questions and Answers – Operator Overloading – 1

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
class A
{
    static int a;
public:
    A()
    {
        cout<<"Object of A is created\n";
    }
    void show()
    {
        a++;
        cout<<"a: "<<a<<endl;
    }
};

class B
{
public:
};

int main(int argc, char const *argv[])
{
    A a1, a2;
    A a3 = a1 + a2;
    return 0;
}
```

- a) Error as a private member a is referenced outside the class
- b) Segmentation fault
- c) No output
- d) Program compiles successfully but gives run-time error

View Answer

Answer: c

Explanation: As every static member must be initialized and we have initialized variable 'a' so no run time error. Also as variable 'a' is a static member and is referenced using the class for initialization therefore no compiler error.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
class A
{
    static int a;
public:
    A()
    {
        cout<<"Object of A is created\n";
    }
    void show()
    {
        a++;
        cout<<"a: "<<a<<endl;
    }
};

class B
{
}
```

```

    public:
};

int main(int argc, char const *argv[])
{
    A a1, a2;
    A a3 = a1 + a2;
    return 0;
}

```

- a) Error because s1+s2 will result into string and no string has substr() function
- b) Segmentation fault as two string cannot be added in C++
- c) The statements runs perfectly
- d) Run-time error

View Answer

Answer: c

Explanation: string is class in C++, therefore when we do (s1+s2) a temporary object is created which stores the result of s1+s2 and then that object calls the function substr() and as that is an object of string class hence substr is a callable function for that temporary string object.

3. What is the output of the following C++ code?

```

#include <iostream>
#include <string>
using namespace std;
class A
{
    static int a;
public:
    A()
    {
        cout<<"Object of A is created\n";
    }
    void show()
    {
        a++;
        cout<<"a: "<<a<<endl;
    }
};

class B
{
public:
};

int main(int argc, char const *argv[])
{
    A a1, a2;
    A a3 = a1 + a2;
    return 0;
}

```

- a) Runs perfectly
- b) Run-time Error
- c) Segmentation fault
- d) Compile-time Error

View Answer

Answer: d

Explanation: As the programmer has not defined what action should be taken when two objects of class A are added, so the program doesn't know and gives compile time error.

3. What is the output of the following C++ code?

```

#include <iostream>
#include <string>

```

```
using namespace std;
class A
{
    static int a;
public:
    A()
    {
        cout<<"Object of A is created\n";
    }
    void show()
    {
        a++;
        cout<<"a: "<<a<<endl;
    }
};

class B
{
public:
};

int main(int argc, char const *argv[])
{
    A a1, a2;
    A a3 = a1 + a2;
    return 0;
}
```

4. What is operator overloading in C++?

- a) Overriding the operator meaning by the user defined meaning for user defined data type
- b) Redefining the way operator works for user defined types
- c) Ability to provide the operators with some special meaning for user defined data type
- d) All of the mentioned

View Answer

Answer: d

Explanation: Operator overloading helps programmer to give his/her own meaning to an operator for user defined data types(eg, classes).

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
class A
{
    static int a;
public:
    A()
    {
        cout<<"Object of A is created\n";
    }
    void show()
    {
        a++;
        cout<<"a: "<<a<<endl;
    }
};

class B
{
public:
};

int main(int argc, char const *argv[])
{
    A a1, a2;
    A a3 = a1 + a2;
    return 0;
}
```

5. What is the syntax of overloading operator + for class A?

- a) A operator+(argument_list){}
- b) A operator[+](argument_list){}
- c) int +(argument_list){}
- d) int [+](argument_list){}

View Answer

Answer: a

Explanation: The general syntax for operator overloading is:

return_type operator_keywordOperator(argument_list){}

eg.

A operator+(argument_list){}

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
class A
{
    static int a;
public:
    A()
    {
        cout<<"Object of A is created\n";
    }
    void show()
    {
        a++;
        cout<<"a: "<<a<<endl;
    }
};

class B
{
public:
};

int main(int argc, char const *argv[])
{
    A a1, a2;
    A a3 = a1 + a2;
    return 0;
}
```

6. How many approaches are used for operator overloading?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: c

Explanation: There are 3 different approaches used for operator overloading:

- i. Overloading unary operator.
- ii. Overloading binary operator.
- iii. Overloading binary operator using a friend function.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
```

```
using namespace std;
class A
{
    static int a;
public:
    A()
    {
        cout<<"Object of A is created\n";
    }
    void show()
    {
        a++;
        cout<<"a: "<<a<<endl;
    }
};

class B
{
public:
};

int main(int argc, char const *argv[])
{
    A a1, a2;
    A a3 = a1 + a2;
    return 0;
}
```

7. Which of the following operator cannot be overloaded?

- a) +
- b) ?:
- c) -
- d) %

View Answer

Answer: b

Explanation: ?:, :: and . cannot be overloaded +, -, % can be overloaded.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
class A
{
    static int a;
public:
    A()
    {
        cout<<"Object of A is created\n";
    }
    void show()
    {
        a++;
        cout<<"a: "<<a<<endl;
    }
};

class B
{
public:
};

int main(int argc, char const *argv[])
{
    A a1, a2;
    A a3 = a1 + a2;
    return 0;
}
```

8. Which of the following operator can be overloaded?

- a) ?:
- b) ::
- c) .
- d) ==

View Answer

Answer: d

Explanation: ?, :: and . cannot be overloaded whereas == can be overloaded.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
class A
{
    static int a;
public:
    A()
    {
        cout<<"Object of A is created\n";
    }
    void show()
    {
        a++;
        cout<<"a: "<<a<<endl;
    }
};

class B
{
public:
};

int main(int argc, char const *argv[])
{
    A a1, a2;
    A a3 = a1 + a2;
    return 0;
}
```

9. Which of the following operator cannot be used to overload when that function is declared as friend function?

- a) -=
- b) ||
- c) ==
- d) []

View Answer

Answer: d

Explanation: When an operator overloaded function is declared as friend function then [] cannot be overloaded.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
class A
{
    static int a;
public:
    A()
    {
        cout<<"Object of A is created\n";
    }
}
```

```

        void show()
        {
            a++;
            cout<<"a: "<<a<<endl;
        }
};

class B
{
    public:
};

int main(int argc, char const *argv[])
{
    A a1, a2;
    A a3 = a1 + a2;
    return 0;
}

```

10. Which of the following operator can be used to overload when that function is declared as friend function?

- a) []
- b) ()
- c) ->
- d) |=

View Answer

Answer: d

Explanation: When an operator overloaded function is declared as friend function then [], () and -> cannot be overloaded.

3. What is the output of the following C++ code?

```

#include <iostream>
#include <string>
using namespace std;
class A
{
    static int a;
    public:
    A()
    {
        cout<<"Object of A is created\n";
    }
    void show()
    {
        a++;
        cout<<"a: "<<a<<endl;
    }
};

class B
{
    public:
};

int main(int argc, char const *argv[])
{
    A a1, a2;
    A a3 = a1 + a2;
    return 0;
}

```

11. In case of non-static member functions how many maximum object arguments a unary operator overloaded function can take?

- a) 1
- b) 2
- c) 3
- d) 0

View Answer

Answer: d

Explanation: In the case of non-static member functions unary operator overloaded function should not take any object argument.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
class A
{
    static int a;
public:
    A()
    {
        cout<<"Object of A is created\n";
    }
    void show()
    {
        a++;
        cout<<"a: "<<a<<endl;
    }
};

class B
{
public:
};

int main(int argc, char const *argv[])
{
    A a1, a2;
    A a3 = a1 + a2;
    return 0;
}
```

12. In case of non-static member functions how many maximum object arguments a binary operator overloaded function can take?

- a) 1
- b) 2
- c) 3
- d) 0

View Answer

Answer: a

Explanation: In the case of non-static member functions binary operator overloaded function should take maximum one object argument only.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
class A
{
    static int a;
public:
    A()
    {
        cout<<"Object of A is created\n";
    }
    void show()
    {
        a++;
        cout<<"a: "<<a<<endl;
    }
};

class B
{
}
```



```

    public:
};

int main(int argc, char const *argv[])
{
    A a1, a2;
    A a3 = a1 + a2;
    return 0;
}

```

13. In the case of friend operator overloaded functions how many maximum object arguments a unary operator overloaded function can take?

- a) 1
- b) 2
- c) 3
- d) 0

View Answer

Answer: a

Explanation: In the case of friend operator overloaded functions unary operator overloaded function should take maximum one object argument only.

3. What is the output of the following C++ code?

```

#include <iostream>
#include <string>
using namespace std;
class A
{
    static int a;
public:
    A()
    {
        cout<<"Object of A is created\n";
    }
    void show()
    {
        a++;
        cout<<"a: "<<a<<endl;
    }
};

class B
{
public:
};

int main(int argc, char const *argv[])
{
    A a1, a2;
    A a3 = a1 + a2;
    return 0;
}

```

14. In the case of friend operator overloaded functions how many maximum object arguments a binary operator overloaded function can take?

- a) 1
- b) 2
- c) 3
- d) 0

View Answer

Answer: b

Explanation: In the case of friend operator overloaded functions binary operator overloaded function should take maximum two object argument.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
class A
{
    static int a;
public:
    A()
    {
        cout<<"Object of A is created\n";
    }
    void show()
    {
        a++;
        cout<<"a: "<<a<<endl;
    }
};

class B
{
public:
};

int main(int argc, char const *argv[])
{
    A a1, a2;
    A a3 = a1 + a2;
    return 0;
}
```

- a) Run-time Error
 - b) Runs perfectly
 - c) Segmentation fault
 - d) Compile-time error
- View Answer

Answer: d

Explanation: .(dot) operator cannot be overloaded therefore the program gives error.

C++ Programming Questions and Answers – Operator Overloading – 2

1. What is a binary operator?

- a) Operator that performs its action on a single operand
- b) Operator that performs its action on two operand
- c) Operator that performs its action on three operand
- d) Operator that performs its action on any number of operands

View Answer

Answer: b

Explanation: As the word binary itself means 2 therefore a binary operator operates on two operands.

2. Which is the correct example of a binary operator?

- a) ++
- b) —
- c) Dereferencing operator(*)
- d) +

View Answer

Answer: d

Explanation: +(adding two operands) requires two operands whereas ++(increases value by 1), -(decreases value by 1) and *(dereferencing operator used for accessing value of pointers) requires only one operand.

3. Which is the correct example of a unary operator?

- a) &
- b) ==
- c) —
- d) /

View Answer

Answer: c

Explanation: &, == and / requires two operands whereas — requires only one operand, in general, it decreases the value of operand by 1.

4. Which is called ternary operator?

- a) ?:
- b) &&
- c) |||
- d) ===

View Answer

Answer: a

Explanation: ?: is called ternary operator because it separates three expressions. exp1 ? exp2 : exp3.

- a) 4 + i6
- b) 2 + i2
- c) Error
- d) Segmentation fault

View Answer

Answer: c

Explanation: In the operator overloaded function we are trying to call default constructor of the class complex but as we have overridden the constructor by our constructor therefore the default constructor cannot be called hence the program gives error.

- a) Complex Number: 4 + i6
- b) Complex Number: 2 + i2

- c) Error
 - d) Segmentation fault
- View Answer

Answer: a

Explanation: As we have defined in the class complex that when we add the two objects of the class complex then add those two complex numbers and show() displays that result.

- a) Complex Number: 4 + i6
 - b) Complex Number: 2 + i2
 - c) Error
 - d) Segmentation fault
- View Answer

Answer: c

Explanation: Each operator function can be defined only once in a class. So as in this program we are trying to define two functions for operator '+' which is not allowed in C++ therefore program gives error.

View Answer

Answer: a

Explanation: As we need to give the result after comparing the capacity of two boxes. We use < operator and as this is the first operand and second operand is passed so we need to do this->capacity < b.capacity (passed object) to make the program run.

9. Which is the correct statement about operator overloading?

- a) Only arithmetic operators can be overloaded
- b) Only non-arithmetic operators can be overloaded
- c) Precedence of operators are changed after overloading
- d) Associativity and precedence of operators does not change

View Answer

Answer: d

Explanation: Both arithmetic and non-arithmetic operators can be overloaded. The precedence and associativity of operators remains the same after and before operator overloading.

10. Pick the incorrect statements out of the following.

- a) Operator overloading does not disturb the precedence of operators
- b) Arity of operators can be changed using operator overloading
- c) No new operators can be created
- d) All of the mentioned

View Answer

Answer: b

Explanation: Arity means a number of operands an operator requires to perform its action and operator overloading does not change the arity of any operator.

- a) Error
 - b) Segmentation fault
 - c) 4
 - d) No output
- View Answer

Answer: a

Explanation: As constructors are defined private and we know objects cannot access private objects therefore program gives error. Also no class should have private constructor.

- a) Error
- b) Segmentation fault

- c) Box 2 has large capacity
- d) No output

View Answer

Answer: a

Explanation: As the operator overloaded function defined is private therefore on comparison the function cannot be called from outside therefore the program gives error.

- a) +
- b) ==
- c) =
- d) ()

View Answer

Answer: b

Explanation: As in the if block we are trying to compare two Box objects and no method is defined to tell compiler how the comparison should be done between these two objects. Hence we need to overload the == operator.

- a) bool operator==();
- b) bool operator==(Box b){}
- c) bool operator==(Box b);
- d) Box operator==();

View Answer

Answer: c

Explanation: In this question we are asked to give the function prototype not the function definition so the answer should not contain {} braces. The correct overloaded function is bool operator==(Box b);

- a) B1's capacity is small
- b) B2's capacity is small
- c) Error
- d) Segmentation fault

View Answer

Answer: b

Explanation: Though the b1's capacity is small the program prints B2's capacity is small because in the < operator overloaded function we are checking B2's capacity < B1's capacity which is false therefore the else is executed.

C++ Programming Questions and Answers – Complex Number Type

1. Which header file is used to declare the complex number?

- a) complexnum
- b) complex
- c) complex number
- d) none of the mentioned

View Answer

Answer: b

Explanation: <complex> header file is used for declaring a complex number in C++.

2. How to declare the complex number?

- a) (3,4)
- b) complex(3,4)
- c) (3,4i)
- d) none of the mentioned

View Answer

Answer: b

Explanation: We can declare the complex number by using complex(3,4) where 3 is a real number and 4 is imaginary part.

3. How many real types are there in complex numbers?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: c

Explanation: There are three real types in complex numbers. They are float complex, double complex, long double complex.

- a) (-240, 128)
- b) (240, 128)
- c) (240, 120)
- d) None of the mentioned

View Answer

Answer: a

Explanation: In this program, we are finding the square of the complex number.

Output:

```
$ g++ comp.cpp
$ a.out
(-240,128)
```

- a) (2, 3)
- b) (4, 5)
- c) (8, 15)
- d) None of the mentioned

View Answer

Answer: b

Explanation: We are just copying the value of c_int into c_double, So it's printing as (4,5).

Output:

```
$ g++ comp1.cpp
$ a.out
(4, 5)
```

- a) (4, 6)
- b) (2, 3)
- c) (6, 12)
- d) None of the mentioned

View Answer

Answer: a

Explanation: We are multiplying the complex number by 2.

Output:

```
$ g++ comp2.cpp
$ a.out
(4, 6)
```

- a) c1: (4,3)(7.65844,6.40819)
- b) c1: (4,3)(7,6)
- c) both c1: (4,3)(7.65844,6.40819) & c1: (4,3)(7,6)
- d) None of the mentioned

View Answer

Answer: a

Explanation: We are adding the two complex numbers and printing the result.

Output:

```
$ g++ comp3.cpp
$ a.out
c1: (4, 3) (7.65844, 6.40819)
```

- a) (4.0, 3.0)
- b) (6.12132, 3.70711)
- c) (5.0, 0.75)
- d) None of the mentioned

View Answer

Answer: b

Explanation: In this program, we are adding both complex number and finding the square root of it.

Output:

```
$ g++ comp4.cpp
$ a.out
(6.12132, 3.70711)
```

9. Which of the following is not a function of complex values?

- a) real
- b) imag
- c) norm
- d) cartesian

View Answer

Answer: d

Explanation: Real is used for returning real part, imag for imaginary part and norm for calculating norm of a complex number. There is no such function Cartesian in complex header file.

- a) 2
- b) 20
- c) 40
- d) None of the mentioned

View Answer

Answer: a

Explanation: `imag` part will return the imaginary part of the complex number.

Output:

```
$ g++ comp5.cpp  
$ a.out  
2
```

ayzom.com

C++ Programming Questions and Answers – Conversion Operators

1. What is the return type of the conversion operator?

- a) void
- b) int
- c) float
- d) no return type

View Answer

Answer: d

Explanation: Conversion operator doesn't have any return type not even void.

2. Why we use the “dynamic_cast” type conversion?

- a) result of the type conversion is a valid
- b) to be used in low memory
- c) result of the type conversion is an invalid
- d) none of the mentioned

View Answer

Answer: a

Explanation: It is used to check that operators and operands are compatible after conversion.

3. How many parameters does a conversion operator may take?

- a) 0
- b) 1
- c) 2
- d) as many as possible

View Answer

Answer: a

Explanation: None.

- a) 20
- b) runtime error
- c) random number
- d) runtime error or random number

View Answer

Answer: d

Explanation: As it assigns to a reference to an object of another incompatible type using explicit type-casting.

Output:

```
$ g++ con.cpp
$ a.out
14032334
```

- a) 3
- b) 4
- c) 5
- d) None of the mentioned

View Answer

Answer: b

Explanation: In this program, we are adding its value with it itself, So only we got the output as 4.

Output:

```
$ g++ con1.cpp
$ a.out
4
```

- a) 5 5
- b) 4 5
- c) 6 6
- d) None of the mentioned

View Answer

Answer: a

Explanation: In this program, we are calculating the magnitude value by two ways.

Output:

```
$ g++ con3.cpp
$ a.out
55
```

- a) converted
- b) error
- c) run time error
- d) none of the mentioned

View Answer

Answer: a

Explanation: In this program, We casted the string to the object of the class.

Output:

```
$ g++ con4.cpp
$ a.out
converted
```

- a) 2110
- b) 1210
- c) 21
- d) None of the mentioned

View Answer

Answer: a

Explanation: In this program, we casted the data type to integer.

Output:

```
$ g++ con5.cpp
$ a.out
2110
```

9. How are types therein user-defined conversion?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: b

Explanation: There are two types of user-defined conversions. They are conversion by the constructor, Conversion functions.

10. Pick out the correct syntax of operator conversion.

- a) operator float()const
- b) operator float()
- c) operator const
- d) none of the mentioned

View Answer

Answer: a

Explanation: None.

ayzom.com

C++ Programming Questions and Answers – Friends

1. Which rule will not affect the friend function?

- a) private and protected members of a class cannot be accessed from outside
- b) private and protected member can be accessed anywhere
- c) protected member can be accessed anywhere
- d) none of the mentioned

View Answer

Answer: a

Explanation: Friend is used to access private and protected members of a class from outside the same class.

2. Which keyword is used to declare the friend function?

- a) firend
- b) friend
- c) classfriend
- d) myfriend

View Answer

Answer: b

Explanation: friend keyword is used to declare a friend function in C++.

3. What is the syntax of friend function?

- a) friend class1 Class2;
- b) friend class;
- c) friend class
- d) none of the mentioned

View Answer

Answer: a

Explanation: In option a, the class2 is the friend of class1 and it can access all the private and protected members of class1.

a) 40

b) 5

c) 10

d) 20

View Answer

Answer: d

Explanation: We are using the friend function for printwidth and multiplied the width value by 2, So we got the output as 20

Output:

```
$ g++ friend.cpp
$ a.out
20
```

a) 20

b) 16

c) 24

d) None of the mentioned

View Answer

Answer: c

Explanation: In this program, we are using the friend function for duplicate function and calculating the area of the rectangle.

Output:

```
$ g++ friend1.cpp
$ a.out
24
```

- a) 24
- b) 35
- c) 16
- d) 36

View Answer

Answer: d

Explanation: In this program, we are using the friend for the class and calculating the area of the square.

Output:

```
$ g++ friend2.cpp
$ a.out
36
```

- a) 200
- b) 150
- c) 100
- d) 300

View Answer

Answer: a

Explanation: In this program, We are finding the mean value by declaring the function mean as a friend of class base.

Output:

```
$ g++ friend3.cpp
$ a.out
200
```

- a) 100
- b) 200
- c) 300
- d) 295

View Answer

Answer: d

Explanation: In this program, we are finding a value from the given function by using the friend for compute function.

Output:

```
$ g++ friend4.cpp
$ a.out
295
```

9. Pick out the correct statement.

- a) A friend function may be a member of another class
- b) A friend function may not be a member of another class
- c) A friend function may or may not be a member of another class
- d) None of the mentioned

View Answer

Answer: c

Explanation: None.

10. Where does keyword 'friend' should be placed?

- a) function declaration
- b) function definition
- c) main function
- d) none of the mentioned

View Answer

Answer: a

Explanation: The keyword friend is placed only in the function declaration of the friend function and not in the function definition because it is used to access the member of a class.

ayzom.com

C++ Programming Questions and Answers – Friend Function

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
class Box
{
    int capacity;
public:
    Box(int cap){
        capacity = cap;
    }
    friend void show();
};

void Box::show()
{
    Box b(10);
    cout<<"Value of capacity is: "<<b.capacity<<endl;
}

int main(int argc, char const *argv[])
{
    show();
    return 0;
}
```

1. What is a friend function in C++?

- a) A function which can access all the private, protected and public members of a class
- b) A function which is not allowed to access any member of any class
- c) A function which is allowed to access public and protected members of a class
- d) A function which is allowed to access only public members of a class

View Answer

Answer: a

Explanation: Friend function in C++ is a function which can access all the private, protected and public members of a class.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
class Box
{
    int capacity;
public:
    Box(int cap){
        capacity = cap;
    }
    friend void show();
};

void Box::show()
{
    Box b(10);
    cout<<"Value of capacity is: "<<b.capacity<<endl;
}

int main(int argc, char const *argv[])
{
    show();
    return 0;
}
```

- a) Value of capacity is: 10
- b) Value of capacity is: 100
- c) Error
- d) Segmentation fault

View Answer

Answer: a

Explanation: As show() is a friend function of class Box hence any object from this function can access the private member of the class Box.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
class Box
{
    int capacity;
public:
    Box(int cap){
        capacity = cap;
    }
    friend void show();
};

void Box::show()
{
    Box b(10);
    cout<<"Value of capacity is: "<<b.capacity<<endl;
}

int main(int argc, char const *argv[])
{
    show();
    return 0;
}
```

- a) Value of capacity is: 10
- b) Value of capacity is: 100
- c) Error
- d) Segmentation fault

View Answer

Answer: c

Explanation: Though it is used to declare the friend functions inside classes they are not members of any class therefore when we giving the definition to friend function show() we should not use Box::show() way of defining it.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
class Box
{
    int capacity;
public:
    Box(int cap){
        capacity = cap;
    }
    friend void show();
};

void Box::show()
{
    Box b(10);
    cout<<"Value of capacity is: "<<b.capacity<<endl;
}
```



```
int main(int argc, char const *argv[])
{
    show();
    return 0;
}
```

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: b

Explanation: A friend functions are not members of any class. Hence this class has only 2 member functions.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
class Box
{
    int capacity;
public:
    Box(int cap){
        capacity = cap;
    }
    friend void show();
};

void Box::show()
{
    Box b(10);
    cout<<"Value of capacity is: "<<b.capacity<<endl;
}

int main(int argc, char const *argv[])
{
    show();
    return 0;
}
```

- a) value of b is: 10
- b) value of b is: 12345435
- c) error
- d) segmentation fault

View Answer

Answer: c

Explanation: There is two error in the program. First the program doesn't have a default constructor for the class B which is used when the object of B is declared inside the class C. Second show() is friend function of class C therefore it can access only private member of class C, not B therefore when we are doing c.b.b here the last b is private member of class B which is not accessible.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
class Box
{
    int capacity;
public:
    Box(int cap){
        capacity = cap;
    }
}
```

```

        friend void show();
};

void Box::show()
{
    Box b(10);
    cout<<"Value of capacity is: "<<b.capacity<<endl;
}

int main(int argc, char const *argv[])
{
    show();
    return 0;
}

```

- a) value of b is: 10
 - b) value of b is: 12345435
 - c) error
 - d) segmentation fault
- View Answer

Answer: a

Explanation: The program follows correct syntax and semantics hence no errors.

3. What is the output of the following C++ code?

```

#include <iostream>
#include <string>
using namespace std;
class Box
{
    int capacity;
public:
    Box(int cap){
        capacity = cap;
    }
    friend void show();
};

void Box::show()
{
    Box b(10);
    cout<<"Value of capacity is: "<<b.capacity<<endl;
}

int main(int argc, char const *argv[])
{
    show();
    return 0;
}

```

- a) value of b is: 10
 - b) value of b is: 12345435
 - c) error
 - d) segmentation fault
- View Answer

Answer: c

Explanation: No function show() is defined in the scope of main() function.

3. What is the output of the following C++ code?

```

#include <iostream>
#include <string>
using namespace std;
class Box

```

```

{
    int capacity;
public:
    Box(int cap){
        capacity = cap;
    }
    friend void show();
};

void Box::show()
{
    Box b(10);
    cout<<"Value of capacity is: "<<b.capacity<<endl;
}

int main(int argc, char const *argv[])
{
    show();
    return 0;
}

```

- a) value of b is: 10
- b) value of b is: 12345435
- c) error
- d) segmentation fault

View Answer

Answer: c

Explanation: Friend functions are not members of any class therefore they should not be called using class objects.

3. What is the output of the following C++ code?

```

#include <iostream>
#include <string>
using namespace std;
class Box
{
    int capacity;
public:
    Box(int cap){
        capacity = cap;
    }
    friend void show();
};

void Box::show()
{
    Box b(10);
    cout<<"Value of capacity is: "<<b.capacity<<endl;
}

int main(int argc, char const *argv[])
{
    show();
    return 0;
}

```

9. Pick the correct statement.

- a) Friend functions are in the scope of a class
- b) Friend functions can be called using class objects
- c) Friend functions can be invoked as a normal function
- d) Friend functions can access only protected members not the private members

View Answer

Answer: c

Explanation: Friend functions are not in the scope of a class and hence cannot be called through a class object. A friend function can access all types of members of the class. They can be invoked as a normal function.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
class Box
{
    int capacity;
public:
    Box(int cap){
        capacity = cap;
    }
    friend void show();
};

void Box::show()
{
    Box b(10);
    cout<<"Value of capacity is: "<<b.capacity<<endl;
}

int main(int argc, char const *argv[])
{
    show();
    return 0;
}
```

10. Which of the following is correct about friend functions?

- a) Friend functions use the dot operator to access members of a class using class objects
- b) Friend functions can be private or public
- c) Friend cannot access the members of the class directly
- d) All of the mentioned

View Answer

Answer: d

Explanation: Friend function can be declared either in private or public part of the class. A friend function cannot access the members of the class directly. They use the dot membership operator with a member name.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
class Box
{
    int capacity;
public:
    Box(int cap){
        capacity = cap;
    }
    friend void show();
};

void Box::show()
{
    Box b(10);
    cout<<"Value of capacity is: "<<b.capacity<<endl;
}

int main(int argc, char const *argv[])
{
    show();
    return 0;
}
```

11. Which keyword is used to represent a friend function?

- a) friend
- b) Friend
- c) friend_func
- d) Friend_func

View Answer

Answer: a

Explanation: friend keyword is used to declare a friend function.

ayzom.com

C++ Programming Questions and Answers – Large Objects

1. How to store the large objects in c++ if it extends its allocated memory?

- a) memory heap
- b) stack
- c) queue
- d) none of the mentioned

View Answer

Answer: a

Explanation: None.

2. When we are using heap operations what do we need to do to save the memory?

- a) rename the objects
- b) delete the objects after processing
- c) both rename & delete the objects
- d) none of the mentioned

View Answer

Answer: b

Explanation: when you allocate memory from the heap, you must remember to clean up objects when you're done! Failure to do so is called a memory leak.

3. Which container in c++ will take large objects?

- a) string
- b) class
- c) vector
- d) none of the mentioned

View Answer

Answer: c

Explanation: Because the vector is mainly used to store large objects for the game programming and other operations etc.

- a) X::operator=(X const &)
- b) X::X(X const &)
- c) X::X()
- d) None of the mentioned

View Answer

Answer: c

Explanation: As we are passing the object without any attributes it will return as X::X().

Output:

```
$ g++ large.cpp
$ a.out
X::X()
```

5. How to stop your program from eating so much ram?

- a) Find a way to work with the data one at a time
- b) Declare it in program memory, instead of on the stack
- c) Use the hard drive, instead of RAM
- d) All of the mentioned

View Answer

Answer: d

Explanation: None.

6. Which option is best to eliminate the memory problem?

- a) use smart pointers
- b) use raw pointers
- c) use virtual destructor
- d) use smart pointers & virtual destructor

View Answer

Answer: d

Explanation: Virtual destructor means is that the object is destructed in reverse order in which it was constructed and the smart pointer will delete the object from memory when the object goes out of scope.

7. What is the size of the heap?

- a) 10MB
- b) 500MB
- c) 1GB
- d) Size of the heap memory is limited by the size of the RAM and the swap memory

View Answer

Answer: d

Explanation: None.

8. How to unlimit the size of the stack?

- a) `setrlimit()`
- b) `unlimit()`
- c) both `setrlimit()` & `unlimit()`
- d) none of the mentioned

View Answer

Answer: a

Explanation: None.

9. In Linux, how do the heaps and stacks are managed?

- a) ram
- b) secondary memory
- c) virtual memory
- d) none of the mentioned

View Answer

Answer: c

Explanation: In virtual memory, We can keep track of all the objects and access them much faster than any another.

10. Which is used to pass the large objects in c++?

- a) pass by value
- b) pass by reference
- c) both pass by value & reference
- d) none of the mentioned

View Answer

Answer: b

Explanation: Because by using pass by reference we need to pass only address location, So it can save a lot of memory.

C++ Programming Questions and Answers – Essential Operators

1. What are the essential operators in c++?

- a) +
- b) |
- c) <=
- d) All of the mentioned

View Answer

Answer: d

Explanation: None.

2. In which direction does the assignment operation will take place?

- a) left to right
- b) right to left
- c) top to bottom
- d) none of the mentioned

View Answer

Answer: b

Explanation: In assignment operation, the flow of execution will be from right to left only.

3. Pick out the compound assignment statement.

- a) $a = a - 5$
- b) $a = a / b$
- c) $a -= 5$
- d) None of the mentioned

View Answer

Answer: c

Explanation: When we want to modify the value of a variable by performing an operation on the value currently stored, We will use compound assignment statement. In this option, $a -= 5$ is equal to $a = a - 5$.

- a) a:4 b:7
- b) a:10 b:4
- c) a:4 b:10
- d) None of the mentioned

View Answer

Answer: a

Explanation: In this program, we are reassigning the values of a and b because of this we got the output as a:4 b:7

Output:

```
$ g++ ess.cpp
$ a.out
a:4 b:7
```

- a) 2
- b) 7
- c) 9
- d) 14

View Answer

Answer: b

Explanation: We are using the ternary operator to evaluate this expression. It will return first option, if first condition is true otherwise it will return second

Output:

```
$ g++ ess1.cpp
$ a.out
7
```

- a) true
- b) false
- c) error
- d) none of the mentioned

View Answer

Answer: b

Explanation: && is called as Logical AND operator, if there is no zero in the operand means, it will be true otherwise false.

Output:

```
$ g++ ess2.cpp
$ a.out
false
```

7. What is the associativity of add(+);

- a) right to left
- b) left to right
- c) both of these
- d) none of the mentioned

View Answer

Answer: b

Explanation: None.

8. What is the name of | operator?

- a) sizeof
- b) or
- c) and
- d) modulus

View Answer

Answer: b

Explanation: | operator is used to find the 'or' of given values.

9. Which operator is having the highest precedence in c++?

- a) array subscript
- b) Scope resolution operator
- c) static_cast
- d) dynamic_cast

View Answer

Answer: b

Explanation: None.

- a) 50
- b) 60
- c) 70
- d) 90

View Answer

Answer: a

Explanation: In this program, the value e is evaluated by precedence and we got the output as 50.

Output:

```
$ g++ ess4.cpp  
$ a.out  
50
```

ayzom.com

C++ Programming Questions and Answers – Function Call

1. What is the use of function call operator?

- a) overloading the methods
- b) overloading the objects
- c) overloading the parameters
- d) none of the mentioned

View Answer

Answer: b

Explanation: None.

2. Pick out the correct statement.

- a) virtual functions does not give the ability to write a templated function
- b) virtual functions does not give the ability to rewrite a templated function
- c) virtual functions does give the ability to write a templated function
- d) none of the mentioned

View Answer

Answer: a

Explanation: None.

3. What will happen when the function call operator is overloaded?

- a) It will not modify the functions
- b) It will modify the functions
- c) It will modify the object
- d) It will modify the operator to be interpreted

View Answer

Answer: d

Explanation: It will modifies how the operator is to be interpreted when applied to objects of a given type.

a) First Distance : 1110

Second Distance :30120

b) First Distance : 110

Second Distance :3020

c) First Distance : 1115

Second Distance :30125

d) none of the mentioned

View Answer

Answer: a

Explanation: We are calculating the foot and inches by using the function call operator.

Output:

```
$ g++ call.cpp
$ a.out
First Distance : 1110
Second Distance :30120
```

a) 1468

b) 2812

c) 2614

d) none of the mentioned

View Answer

Answer: c

Explanation: We are passing the values by reference and modified the data on the function block.

Output:

```
$ g++ call11.cpp
$ a.out
2614
```

- a) 55, 106, 156
 - b) 55, 106
 - c) 55, 106, 159
 - d) none of the mentioned
- View Answer

Answer: a

Explanation: In this program, We are using the function call operator to calculate the value of objc.

Output:

```
$ g++ call12.cpp
$ a.out
55, 106, 156
```

- a) c2=(9.7,8)
 - c2=(5.1,5.3)
 - b) c2=(9,8)
 - c2=(5,5)
 - c) c2=(4.7,8)
 - c2=(2.1,5.3)
 - d) none of the mentioned
- View Answer

Answer: a

Explanation: In this program, We are returning the real and imaginary part of the complex number by using function call operator.

Output:

advertisement

```
$ g++ call13.cpp
$ a.out
c2=(9.7,8)
c2=(5.1,5.3)
```

8. In which form does the function call operator can be overloaded?

- a) static member function
 - b) non-static member function
 - c) dynamis_cast
 - d) static_cast
- View Answer

Answer: b

Explanation: None.

- a) 119
 - b) 102.5
 - c) 123.4
 - d) none of the mentioned
- View Answer

Answer: b

Explanation: In this program, We are overloading the function and getting the output as 10 and 2.5 by division and multiplication.

Output:

```
$ g++ call3.cpp
$ a.out
102.5
```

10. What is the use of functor?

- a) It makes the object "callable" like a function
- b) It makes the class "callable" like a function
- c) It makes the attribute "callable" like a function
- d) None of the mentioned

View Answer

Answer: a

Explanation: None.

ayzom.com

C++ Programming Questions and Answers – Dereferencing

1. Which is used to tell the computer that where a pointer is pointing to?

- a) dereference
- b) reference
- c) heap operations
- d) none of the mentioned

View Answer

Answer: a

Explanation: None.

2. Which is used to do the dereferencing?

- a) pointer without asterix
- b) value without asterix
- c) pointer with asterix
- d) value with asterix

View Answer

Answer: c

Explanation: Dereferencing is using a pointer with asterix. For example, *(abc).

3. Pick out the correct option.

- a) References automatically dereference without needing an extra character
- b) References automatically dereference with an extra character
- c) Reference will not dereference
- d) None of the mentioned

View Answer

Answer: a

Explanation: None.

a) 100 200

b) 100 0

c) 200 200

d) 100 100

View Answer

Answer: d

Explanation: In this program, We are making the assignments and invoking the both b and c values as 100 by dereference operator.

Output:

```
$ g++ def.cpp
$ a.out
100 100
```

a) 5

b) 10

c) memory address

d) none of the mentioned

View Answer

Answer: a

Explanation: In this program, we are copying the memory location of x into p and then printing the value in the address.

Output:

```
$ g++ def1.cpp  
$ a.out  
5
```

- a) 1
1
0xbffc9924
1
b) 1
1
1
0xbffc9924
c) 1
0xbffc9924
1
&1
d) none of the mentioned
View Answer

Answer: a
Explanation: In this program, We are printing the values and memory address by using the pointer and dereference operator.
Output:

```
$ g++ def2.cpp  
$ a.out  
1  
1  
0xbffc9924  
1
```

- a) 4
b) 2
c) Depends on compiler
d) None of the mentioned
View Answer

Answer: c
Explanation: The size of a data type mainly depends on compiler only.
Output:

advertisement

```
$ g++ def3.cpp  
$ a.out  
4
```

- a) 7
0xbff99fc98
8
5
14
b) 7
8
0xbff99fc98
5
14
c) 0xbff99fc98
d) none of the mentioned

View Answer

Answer: a

Explanation: In this program, We are printing the values that are pointed by pointer and also the dereference operator.

Output:

```
$ g++ def5.cpp
$ a.out
7
0xbf99fc98
8
5
14
```

9. What does the dereference operator will return?

- a) rvalue equivalent to the value at the pointer address
- b) lvalue equivalent to the value at the pointer address
- c) it will return nothing
- d) none of the mentioned

View Answer

Answer: b

Explanation: It operates on a pointer variable, and returns an l-value equivalent to the value at the pointer address.

10. Pick out the correct statement.

- a) The NULL pointer dereference occurs where a pointer that is expected to be a valid address but instead is equal to NULL
- b) The NULL pointer dereference occurs where a pointer that is expected to be a valid address but instead is equal to the memory address
- c) All of the mentioned
- d) None of the mentioned

View Answer

Answer: a

Explanation: None.

C++ Programming Questions and Answers – Increment and Decrement

1. Which operator works only with integer variables?

- a) increment
- b) decrement
- c) both increment & decrement
- d) none of the mentioned

View Answer

Answer: c

Explanation: Because increment and decrement operator increases and decreases values of values and no such things define in strings so cannot be used with strings. Also they cannot be used with floats and doubles because there is no way to fix how much the value should be increased or decreased if increment or decrement operator is applied on such variables. That's why both these operators only works with integer values.

2. How many types are there in increment/decrement operator?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: b

Explanation: There are two types of increment/decrement. They are postfix and prefix.

3. Pick out the correct statement.

- a) Increment operator ++ adds 1 to its operand
- b) Increment operator ++ adds 2 to its operand
- c) Decrement operator -- subtracts 1 to its operand
- d) None of the mentioned

View Answer

Answer: a

Explanation: Increment operator are used to increase the values of any integer variable by 1.

a) 21

b) 22

c) 23

d) 20

View Answer

Answer: a

Explanation: value of 'a' will be stored in c and then only it will be incremented.

Output:

```
$ g++ incre.cpp
$ a.out
21
```

a) 55

b) 64

c) 46

d) 45

View Answer

Answer: b

Explanation: The values will be pre increment and pre decrement, So it will print as 64.

Output:

```
$ g++ incre2.cpp
$ a.out
64
```

- a) 10
- b) 11
- c) 9
- d) 12

View Answer

Answer: a

Explanation: In this program, the increment and decrement of evaluation of z will not be accounted because they are post.

Output:

```
$ g++ incre3.cpp
$ a.out
10
```

- a) 11
- b) 12
- c) 13
- d) 14

View Answer

Answer: d

Explanation: In this program, we are adding the x value after pre incrementing two times.

Output:

advertisement

```
$ g++ incre4.cpp
$ a.out
14
```

- a) 532
- b) 235
- c) 312
- d) 311

View Answer

Answer: d

Explanation: In this program, We are pre increment and post incrementing the operands and saving it.

Output:

```
$ g++ incre5.cpp
$ a.out
311
```

9. Pick out the correct statement

- a) Pre Increment is faster than post-increment
- b) post-increment is faster than Pre Increment
- c) pre increment is slower than post-increment
- d) none of the mentioned

View Answer

Answer: a

Explanation: Because Pre Increment take one-byte instruction & post increment takes two-byte instruction.

10. Which concepts does the Pre Increment use?

- a) call by value
- b) call by reference
- c) queue
- d) none of the mentioned

View Answer

Answer: b

Explanation: call by reference because the changes are reflected back to the same memory cells/variables.

ayzom.com

C++ Programming Questions and Answers – String Class

1. How many types of representation are in the string?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: b

Explanation: C++ provides following two types of string representations. They are C-style character string and string class type with Standard C++.

View Answer

Answer: c

Explanation: `#include<string>` is the header file for the string class.

3. Which is used to return the number of characters in the string?

- a) length
- b) size
- c) both size & length
- d) none of the mentioned

View Answer

Answer: c

Explanation: Both will return the number of characters that conform to the string's content.

- a) 5
- b) 55
- c) 11
- d) 10

View Answer

Answer: d

Explanation: In the program, We are concatenating the str1 and str2 and printing it's total length. So the length is 10.

Output:

```
$ g++ stri.cpp
$ a.out
10
```

- a) microsoft
- b) micro
- c) tfosorcim
- d) tfos

View Answer

Answer: c

Explanation: 'rbegin' is used to reverse the given the string.

Output:

```
$ g++ stril.cpp
$ a.out
tfosorcim
```

- a) nobody does like this
- b) nobody
- c) everybody
- d) everybody does like this

View Answer

Answer: d

Explanation: rfind is used to find the characters in the string and replace is used to replace with certain characters.

Output:

```
$ g++ stri2.cpp
$ a.out
everybody does like this
```

- a) jobs is
- b) steve legend
- c) steve
- d) none of the mentioned

View Answer

Answer: b

Explanation: In this program, We are leaving the first 5 characters and last 7 characters and we are erasing the remaining the characters.

Output:

```
$ g++ stri3.cpp
$ a.out
steve legend
```

- a) M
- b) Microsoft
- c) Micro
- d) runtime error

View Answer

Answer: d

Explanation: This program will terminate because the cout element is out of range.

View Answer

Answer: d

Explanation: str.capacity() returns the size of the storage space currently allocated for the string, expressed in terms of bytes and capacity of the string may be equal or greater. The max_size() returns the max size of the string.

Output:

```
$ g++ stri5.cpp
$ a.out
15
9223372036854775807
```

10. Which method do we use to append more than one character at a time?

- a) append
- b) operator+=
- c) data
- d) both append & operator+=

View Answer

Answer: d

Explanation: C++ allows to append more characters to string using both inbuilt append() function and using operator overloaded += operator.

C++ Programming Questions and Answers – String – 1

1. What is string objects in C++?

- a) Stream of alphabets
- b) A stream of well-defined characters
- c) Stream of characters
- d) A stream of characters terminated by \0

View Answer

Answer: b

Explanation: String is defined as streams of characters, not necessarily terminated by \0. Also, a string can contain characters other than alphabets.

2. What is Character-Array?

- a) array of alphabets
- b) array of well-defined characters
- c) array of characters
- d) array of characters terminated by \0

View Answer

Answer: c

Explanation: Character-Array is defined as an array of characters, not necessarily terminated by \0. Also, a character-array can contain characters other than alphabets.

3. Pick the incorrect statement about Character-Array.

- a) Character-Array can be terminated by a null character('\0')
- b) Character-Array has a static size
- c) Character-Array has a dynamic size
- d) Character-Array has a threat of array-decay

View Answer

Answer: c

Explanation: As Character-Array is an array, its size should be defined during its declaration hence the size of Character-Array is static. A Character-Array is not necessarily to be terminated by a null character. Also, it has a threat of array-decay.

4. Pick the correct statement about string objects in C++.

- a) String objects must be terminated by a null character('\0')
- b) String objects have a static size
- c) String objects have a dynamic size
- d) String objects use extra memory than required.

View Answer

Answer: c

Explanation: String objects are dynamic in nature i.e. their size varies as their value changes so they don't use any extra memory and it is not necessary to terminate a string object by '\0'.

- a) str
- b) Input provided by the user
- c) Error
- d) Segmentation fault

View Answer

Answer: b

Explanation: There is no error in the program and as we are asking the user to enter a string and printing that string to console. Therefore output will be the string provided by the user.

- a) H
- b) e
- c) Error
- d) o

View Answer

Answer: a

Explanation: The program has no errors so and as str = "Hello World" and we are trying to print the first character of str. Hence "H" is the answer.

-
- a) Compiler-time Error
 - b) Run-time Error
 - c) Input given by the user
 - d) Depends on the length of the string entered by the user

View Answer

Answer: d

Explanation: As the character array size is 10 so if the string entered by the user is ≤ 10 then there will be no error and the program runs perfectly otherwise if the length is > 10 then the program gives a run-time error because the string crosses the allocated memory space.

Output:

```
length < 10
```

```
$ ./a.out
Hello
Hello
```

```
length > 10
$ ./a.out
C++Programming
*** stack smashing detected ***: terminated
Aborted (core dumped)
```

-
- a) Hello World
 - b) Hello
 - c) World
 - d) Error

View Answer

Answer: b

Explanation: As cin considers \n or space as the terminating symbols for the input so when the user enters "Hello World" so only "Hello" will be stored into the str variable as cin stops scanning input after space.

Output:

```
$ ./a.out
Hello World
Hello
```

9. Which header file is used to include the string object functions in C++?

- a) `#include <string.h>`
- b) `#include <cstring>`
- c) `#include <string>`
- d) `#include <string.cpp>`

View Answer

Answer: c

Explanation: `#include <string>` header file is used as it contains all the string object functions.

-
- a) Hello World
 - b) Hello

- c) World
 - d) Error
- View Answer

Answer: d

Explanation: There is no operation defined for the addition of character array in C++ hence the compiler throws an error as it does not understands what to do about this expression.

- a) Hello World
 - b) Hello
 - c) World
 - d) Error
- View Answer

Answer: a

Explanation: The program runs perfectly as string class has defined the addition of two strings so when two strings are added then both the strings are concatenated. Hence the output is "Hello World".

- a) 1 and 2
 - b) 2 and 3
 - c) 1 and 3
 - d) 1, 2 and 3
- View Answer

Answer: a

Explanation: To concatenate two string objects we are provided with either direct addition or append() function in string class but strcat() is char* function hence they cannot be used to concatenate two string objects.

13. Which of the following is not a modifier function in string class?

- a) operator+=()
- b) operator[]()
- c) push_back()
- d) erase()

View Answer

Answer: b

Explanation: [] operator is used to access one of the characters of the string objects whereas other functions are used to modify the string in some way.

14. Which function is used to get the length of a string object?

- a) str.length()
- b) str.size()
- c) str.max_size()
- d) both size() and length() function

View Answer

Answer: d

Explanation: Both size() and length() are used to get the size of the string objects.

15. What is the identifier given to string class to declare string objects?

- a) String
- b) string
- c) STRING
- d) Any of the above can be used

View Answer

Answer: b

Explanation: string identifier is used as the name of the class string.

ayzom.com

C++ Programming Questions and Answers – String – 2

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
#include <cstring>
using namespace std;
int main()
{
    string s('a');
    cout<<s;
    return 0;
}
```

- a) Hello World
- b) Hello
- c) World
- d) Error

[View Answer](#)

Answer: b

Explanation: char* are terminated by a '\0' character so the string "Hello\0World" will be cut down to "Hello".

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
#include <cstring>
using namespace std;
int main()
{
    string s('a');
    cout<<s;
    return 0;
}
```

- a) a
- b) empty string
- c) Error
- d) Segmentation fault

[View Answer](#)

Answer: a

Explanation: string class has a constructor for this call hence the string s will be assigned "a".

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
#include <cstring>
using namespace std;
int main()
{
    string s('a');
    cout<<s;
    return 0;
}
```

- a) a
- b) empty string
- c) Error
- d) Segmentation fault

View Answer

Answer: c

Explanation: The string class provides string(string s) as a constructor not the string(char) as a constructor therefore this assignment is not valid.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
#include <cstring>
using namespace std;
int main()
{
    string s('a');
    cout<<s;
    return 0;
}
```

- a) 1 only
- b) 2 only
- c) both of them
- d) both are wrong

View Answer

Answer: c

Explanation: string class provides the addition of char and string and also push_back(char) function to append a character at the end of a string.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
#include <cstring>
using namespace std;
int main()
{
    string s('a');
    cout<<s;
    return 0;
}
```

- a) Sanfoundry.!
- b) Sanfoundry.
- c) Sanfoundry!
- d) Sanfoundry!.

View Answer

Answer: c

Explanation: back() function modifies the last character of the string with the character provided.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
#include <cstring>
using namespace std;
int main()
{
    string s('a');
    cout<<s;
    return 0;
}
```

- a) Sanfoundry
- b) Sanfoundry.

- c) sanfoundry
- d) sanfoundry.

View Answer

Answer: b

Explanation: front() modifies the first character of the string with the character provided.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
#include <cstring>
using namespace std;
int main()
{
    string s('a');
    cout<<s;
    return 0;
}
```

- a) foundry.
- b) dry.
- c) oundry.
- d) found

View Answer

Answer: b

Explanation: As we are first taking the substring of s from 3 to end then on that substring we are taking substr from 4 to end which is equal to "dry."

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
#include <cstring>
using namespace std;
int main()
{
    string s('a');
    cout<<s;
    return 0;
}
```

- a) 1511
- b) 1111
- c) 1115
- d) None

View Answer

Answer: a

Explanation: Capacity of a string object is defined as the length of string plus the extra space given to that object which will be used further if string is expanded.

C++ Programming Questions and Answers – Constructors and Destructors – 1

1. What is the role of a constructor in classes?

- a) To modify the data whenever required
- b) To destroy an object
- c) To initialize the data members of an object when it is created
- d) To call private functions from the outer world

View Answer

Answer: c

Explanation: A constructor is used in classes to initialize data members of class in order to avoid errors/segmentation faults.

2. Why constructors are efficient instead of a function `init()` defined by the user to initialize the data members of an object?

- a) Because user may forget to call `init()` using that object leading segmentation fault
- b) Because user may call `init()` more than once which leads to overwriting values
- c) Because user may forget to define `init()` function
- d) All of the mentioned

View Answer

Answer: d

Explanation: We cannot use `init()` because as mentioned in options that user may forget to initialize the members which will lead to a segmentation fault. Also if the user calls the `init()` function more than once it may overwrite the values and may result into disastrous results. Also if any user forgets to define `init()` function then no object will be initialized whereas if any constructor is not defined in any class the class provides a default constructor for initialization.

3. What is a copy constructor?

- a) A constructor that allows a user to move data from one object to another
- b) A constructor to initialize an object with the values of another object
- c) A constructor to check the whether to objects are equal or not
- d) A constructor to kill other copies of a given object.

View Answer

Answer: b

EXplanation: Copy constructor allows the user to initialize an object with the values of another object instead of supplying the same set of values again to initialize the object.

- a) 5
- b) 55
- c) Error
- d) Segmentation Fault

View Answer

Answer: c

Explanation: As we have defined a constructor which takes an `int` parameter, so when we are trying to declare an object `obj` of class `A` without supplying any parameter then as a constructor is overwritten it will give an error saying that no matching function found. So whenever one writes a constructor then the default constructor is overwritten hence if you want to declare an object without parameter then you also have to define that constructor.

- a) 5
- b) 10
- c) Error
- d) Segmentation fault

View Answer

Answer: c

Explanation: Here the constructor is made private and as no object can access any private object directly therefore the program will give error. One should always define a constructor as public.

- a) 3
- b) 4
- c) 2
- d) 1

[View Answer](#)

Answer: d

Explanation: As the object is defined only once in the program at line A B::a, so the constructor of A is called only once. For objects a1, a2 and a3 copy constructor is called so the string will not be printed for them.

7. What happens if a user forgets to define a constructor inside a class?

- a) Error occurs
- b) Segmentation fault
- c) Objects are not created properly
- d) Compiler provides a default constructor to avoid faults/errors

[View Answer](#)

Answer: d

Explanation: The C++ compiler always provides a default constructor if one forgets to define a constructor inside a class.

8. How many parameters does a default constructor require?

- a) 1
- b) 2
- c) 0
- d) 3

[View Answer](#)

Answer: c

Explanation: A default constructor does not require any parameters for object creation that's why sometimes we declare an object without any parameters.

9. How constructors are different from other member functions of the class?

- a) Constructor has the same name as the class itself
- b) Constructors do not return anything
- c) Constructors are automatically called when an object is created
- d) All of the mentioned

[View Answer](#)

Answer: d

Explanation: All the above mention are the reasons where constructor differs from other normal member functions of a class.

10. How many types of constructors are there in C++?

- a) 1
- b) 2
- c) 3
- d) 4

[View Answer](#)

Answer: c

Explanation: There are three types of constructors in C++ namely default, parameterized and copy constructor.

[View Answer](#)

Answer: b

Explanation: When object obj is declared then the default constructor is called. When we are declaring the a1 object as we are equating obj to

a1 object obj will be copied to a1 hence copy constructor is called, similarly when object a2 is created obj is passed as a parameter to function which is available as copy constructor function, hence copy constructor will be called. So one time Default constructor and two times copy constructor.

View Answer

Answer: c

Explanation: Here when we are declaring the object b1 of class B then first the constructor of class B will be called, in which first it will initialize all the members of class B and as obj from class A is member of class B and it should be initialized so the A's default constructor will be called and terminates after that B's constructor terminates hence A's default constructor called is printed before B's constructor called.

13. What is the role of destructors in Classes?

- a) To modify the data whenever required
- b) To destroy an object when the lifetime of an object ends
- c) To initialize the data members of an object when it is created
- d) To call private functions from the outer world

View Answer

Answer: b

Explanation: Destructors are used in Classes to destroy an object after its lifetime is over i.e. to free resources occupied by that object.

14. What is syntax of defining a destructor of class A?

- a) A(){}
- b) ~A(){}
- c) A::A(){}
- d) ~A(){};

View Answer

Answer: b

Explanation: A destructor starts with a ~(tilde) symbol, has the same name as the class.

15. When destructors are called?

- a) When a program ends
- b) When a function ends
- c) When a delete operator is used
- d) All of the mentioned

View Answer

Answer: d

Explanation: Destructors are called at the following time:

- i) at the end of the program to destroy objects declared in the main() or global scope.
- ii) at the end of a function to destroy objects declared at that function scope.
- iii) when user by himself tries to delete an object using the delete operator.
- iv) at the end of a block to destroy objects declared at that block scope.

C++ Programming Questions and Answers – Constructors and Destructors – 2

1. What is the difference between constructors and destructors?

- a) They have a different function name
- b) Constructors does not have return type whereas destructors do have
- c) Constructors allow function parameters whereas destructors do not
- d) All of the mentioned

View Answer

Answer: c

Explanation: Both the constructors and destructors have the same function name and both of them do not have return type but constructors allow function parameters whereas destructors do not.

2. How many Destructors are allowed in a Class?

- a) 1
- b) 2
- c) 3
- d) Any number

View Answer

Answer: a

Explanation: A class in C++ allows only one destructor, which is called whenever the lifetime of an object ends.

View Answer

Answer: c

Explanation: The destructors for an object is called before the destructor of its data members or bases.

View Answer

Answer: c

Explanation: Though B class have no data member of the class but as class B is derived from class A, the destructor of class A will be called to destroy the data inherited from class A to class B.

View Answer

Answer: c

Explanation: There are two calls to constructor of class A, one is for the data member of class B and second because class B is derived from class A. Similarly two destructor calls.

View Answer

Answer: b

Explanation: Here as 'a' is a static member of class B and as all static members should be initialized separately as no object creation initializes static member and as 'a' is not initialized, hence no call will be made to the constructor of class A.

View Answer

Answer: a

Explanation: Here as no object of B is declared so no call to B's constructor but as we have initialised the static member 'a' of class B, hence A's constructor and destructor will be called once.

- a) A a(5);
- b) A a;
- c) A a = A(5);
- d) A a = A();

View Answer

Answer: c

Explanation: Explicit call represents the programmer by himself mentioning the type name. So A a = A(5); is the correct explicit call as we are mentioning typename A(5) from our side, whereas A a = A(); is not the correct call because no such constructor is there in class A.

C++ Programming Questions and Answers – Constructors and Destructors – 3

3. What is the output of following C++ code?

```
#include <iostream>
using namespace std;
class A{
    A() {
        cout<<"Constructor called";
    }
};
int main(int argc, char const *argv[])
{
    A a;
    return 0;
}
```

1. Which of the following constructors are provided by the C++ compiler if not defined in a class?

- a) Default constructor
- b) Assignment constructor
- c) Copy constructor
- d) All of the mentioned

View Answer

Answer: d

Explanation: If a programmer does not define the above constructors in a class the C++ compiler by default provides these constructors to avoid error on basic operations.

3. What is the output of following C++ code?

```
#include <iostream>
using namespace std;
class A{
    A() {
        cout<<"Constructor called";
    }
};
int main(int argc, char const *argv[])
{
    A a;
    return 0;
}
```

2. When a copy constructor is called?

- a) When an object of the class is returned by value
- b) When an object of the class is passed by value to a function
- c) When an object is constructed based on another object of the same class
- d) All of the mentioned

View Answer

Answer: d

Explanation: Copy constructor is called in all the above-mentioned criteria because in all the above cases we are somehow trying to copy one object into another.

3. What is the output of following C++ code?

```
#include <iostream>
using namespace std;
class A{
    A() {
        cout<<"Constructor called";
    }
};
```

```
int main(int argc, char const *argv[])
{
    A a;
    return 0;
}
```

- a) Constructor called
 - b) Nothing printed
 - c) Error
 - d) Segmentation fault
- View Answer

Answer: c

Explanation: No constructor should be made private because objects need to call them and as by default all the members of a class are private therefore constructor defined in the above program is private which is wrong therefore the compiler gives the error.

3. What is the output of following C++ code?

```
#include <iostream>
using namespace std;
class A{
    A() {
        cout<<"Constructor called";
    }
};
int main(int argc, char const *argv[])
{
    A a;
    return 0;
}
```

- a) Constructor called
 - b) Nothing printed
 - c) Error
 - d) Segmentation fault
- View Answer

Answer: b

Explanation: As we have declared a pointer variable for class A but we have not initialized the memory to that pointer and until the memory is not initialized the constructor for the pointer variable will not be called hence nothing is printed on the screen.

3. What is the output of following C++ code?

```
#include <iostream>
using namespace std;
class A{
    A() {
        cout<<"Constructor called";
    }
};
int main(int argc, char const *argv[])
{
    A a;
    return 0;
}
```

- a) 1010
 - b) 87368746
 - c) Error
 - d) Segmentation fault
- View Answer

Answer: a

Explanation: Although the declaration of object a1 looks erroneous but actually it is acceptable by the C++ compiler to take values for class

objects as mentioned above. Next value of a1 is copied to a2 hence 1010 is printed.

Output:

\$./a.out

1010

3. What is the output of following C++ code?

```
#include <iostream>
using namespace std;
class A{
    A() {
        cout<<"Constructor called";
    }
};
int main(int argc, char const *argv[])
{
    A a;
    return 0;
}
```

- a) 10
- b) Compile time error
- c) Run-time error
- d) No output

View Answer

Answer: b

Explanation: The declaration of object a1 needs a constructor without any arguments which is not available in the class as we have overwritten the default constructor, therefore, the program gives the error.

3. What is the output of following C++ code?

```
#include <iostream>
using namespace std;
class A{
    A() {
        cout<<"Constructor called";
    }
};
int main(int argc, char const *argv[])
{
    A a;
    return 0;
}
```

- a) 010
 - b) 100
 - c) 001
 - d) Error
- View Answer

Answer: a

Explanation: As constructor is accepting default parameter therefore the declaration of a1 and a2 both are valid hence the program runs successfully.

3. What is the output of following C++ code?

```
#include <iostream>
using namespace std;
class A{
    A() {
        cout<<"Constructor called";
    }
};
```

```
int main(int argc, char const *argv[])
{
    A a;
    return 0;
}
```

- a) Constructor called
 - b) Nothing printed
 - c) Error
 - d) Segmentation fault
- View Answer

Answer: b

Explanation: Unlike new malloc never calls the constructor of a class hence when we are assigning memory to an object of class A using malloc constructor is not called hence nothing is printed.

3. What is the output of following C++ code?

```
#include <iostream>
using namespace std;
class A{
    A() {
        cout<<"Constructor called";
    }
};
int main(int argc, char const *argv[])
{
    A a;
    return 0;
}
```

- a) Constructor called
 - b) Nothing printed
 - c) Error
 - d) Segmentation fault
- View Answer

Answer: a

Explanation: In this program, we have defined a global variable an outside main function for which constructor will be called hence the output is printed.

3. What is the output of following C++ code?

```
#include <iostream>
using namespace std;
class A{
    A() {
        cout<<"Constructor called";
    }
};
int main(int argc, char const *argv[])
{
    A a;
    return 0;
}
```

- c) Error
 - d) Segmentation fault
- View Answer

Answer: a

Explanation: In this program, we have defined a global variable an outside main function for which constructor will be called. Now as a is a global variable, therefore, the call for constructor will be before the main function hence constructor called will be printed before the Main function.

3. What is the output of following C++ code?

```
#include <iostream>
using namespace std;
class A{
    A() {
        cout<<"Constructor called";
    }
};
int main(int argc, char const *argv[])
{
    A a;
    return 0;
}
```

c) Error

d) Segmentation fault

View Answer

Answer: c

Explanation: As constructor of class A is defined private and we are trying to define an object of class A which cannot call this constructor as it is private therefore the program gives an error.

3. What is the output of following C++ code?

```
#include <iostream>
using namespace std;
class A{
    A() {
        cout<<"Constructor called";
    }
};
int main(int argc, char const *argv[])
{
    A a;
    return 0;
}
```

c) Error

d) Segmentation fault

View Answer

Answer: a

Explanation: Now still the constructor of a class is private but class B is friend class of A hence it can access the private members of class A and as in the above program we defining an object of class A in class B only, therefore, the program runs fine.

3. What is the output of following C++ code?

```
#include <iostream>
using namespace std;
class A{
    A() {
        cout<<"Constructor called";
    }
};
int main(int argc, char const *argv[])
{
    A a;
    return 0;
}
```

c) Error

d) Segmentation fault

View Answer

Answer: c

Explanation: In this program, the destructor of class A is private therefore the destructor for object a cannot be called hence the program gives an error.

3. What is the output of following C++ code?

```
#include <iostream>
using namespace std;
class A{
    A() {
        cout<<"Constructor called";
    }
};
int main(int argc, char const *argv[])
{
    A a;
    return 0;
}
```

14. How destructor overloading is done?

- a) By changing the number of parameters
- b) By changing the parameters type
- c) By changing both the number of parameters and their type
- d) No chance for destructor overloading

View Answer

Answer: d

Explanation: A class is allowed to have only one destructor. Therefore there is no point of destructor overloading.

3. What is the output of following C++ code?

```
#include <iostream>
using namespace std;
class A{
    A() {
        cout<<"Constructor called";
    }
};
int main(int argc, char const *argv[])
{
    A a;
    return 0;
}
```

15. Which of the following is correct?

- a) Destructors can be virtual
- b) There can be more than one destructor in a class
- c) Destructor definition starts with !
- d) Destructor is used to initialize objects

View Answer

Answer: a

Explanation: Destructors can be virtual. They are used to destroy objects. Only one destructor is allowed per class. Destructor definition starts with a tilde(~).

C++ Programming Questions and Answers – Derived Classes

1. Where is the derived class is derived from?

- a) derived
- b) base
- c) both derived & base
- d) None of the mentioned

View Answer

Answer: b

Explanation: Because derived inherits functions and variables from base.

2. Pick out the correct statement.

- a) A derived class's constructor cannot explicitly invokes its base class's constructor
- b) A derived class's destructor cannot invoke its base class's destructor
- c) A derived class's destructor can invoke its base class's destructor
- d) None of the mentioned

View Answer

Answer: b

Explanation: Destructors are automatically invoked when an object goes out of scope or when a dynamically allocated object is deleted. Inheritance does not change this behavior. This is the reason a derived destructor cannot invoke its base class destructor.

3. Which of the following can derived class inherit?

- a) members
- b) functions
- c) both members & functions
- d) none of the mentioned

View Answer

Answer: c

Explanation: Both data members and member functions are inherited by derived class in C++.

- a) 54.3R
- b) R4.35
- c) 4.3R5
- d) None of the mentioned

View Answer

Answer: a

Explanation: In this program, We are passing the value and manipulating by using the derived class.

Output:

```
$ g++ der.cpp
$ a.out
54.3R
```

- a) 3 4
- b) 4 3
- c) 4
- d) 3

View Answer

Answer: b

Explanation: In this program, We are passing the values and assigning it to i and j and we are printing it.

Output:

```
$ g++ der1.cpp
$ a.out
4 3
```

- a) Instantiating Base
Base
Instantiating Derived
Base
Derived
b) Instantiating Base
Instantiating Derived
Base
Derived
c) Instantiating Base
Base
Instantiating Derived
Base
d) None of the mentioned
View Answer

Answer: a

Explanation: In this program, We are printing the execution order of the program.

Output:

```
$ g++ der2.cpp
$ a.out
Instantiating Base
Base
Instantiating Derived
Base
Derived
```

- a) Parent()
Parent()
Child2()
Parent(42)
Child2(42)
~Child2()
~Parent()
~Child2()
~Parent()
~Parent()
b) error
c) runtime error
d) none of the mentioned
View Answer

Answer: b

Explanation: In this program, We got an error in overloading because we didn't invoke the destructor of parent.

- a) 10 100
b) 100 10
c) 10 10
d) 100 100
View Answer

Answer: a

Explanation: In this program, We are passing the value and getting the result by derived class.

Output:

```
$ g++ der5.cpp
$ a.out
10 100
```

9. Which operator is used to declare the destructor?

- a) #
- b) ~
- c) @
- d) \$

View Answer

Answer: b

Explanation: tilde(~) is used to declare destructor of a class.

10. Which constructor will initialize the base class data member?

- a) derived class
- b) base class
- c) class
- d) none of the mentioned

View Answer

Answer: b

Explanation: Because it is having the proper data set to initialize, Otherwise it will throw a error.

ayzom.com

C++ Programming Questions and Answers – Abstract Classes – 2

1. What is an abstract class in C++?

- a) Class specifically used as a base class with atleast one virtual functions
- b) Class specifically used as a base class with atleast one pure virtual functions
- c) Class from which any class is derived
- d) Any Class in C++ is an abstract class

View Answer

Answer: b

Explanation: An abstract class is defined as a class which is specifically used as a base class. An abstract class should have atleast one pure virtual function.

2. What is a pure virtual function in C++?

- a) A virtual function defined in a base class
- b) A virtual function declared in a base class
- c) Any function in a class
- d) A function without definition in a base class

View Answer

Answer: b

Explanation: Pure virtual function is a virtual function which has no definition/implementation in the base class.

3. Which is the correct syntax of defining a pure virtual function?

- a) pure virtual return_type func();
- b) virtual return_type func() pure;
- c) virtual return_type func() = 0;
- d) virtual return_type func();

View Answer

Answer: c

Explanation: virtual return_type function_name(parameters) = 0; where {=0} is called pure specifier.

4. Which is the correct statement about pure virtual functions?

- a) They should be defined inside a base class
- b) Pure keyword should be used to declare a pure virtual function
- c) Pure virtual function is implemented in derived classes
- d) All of the mentioned

View Answer

Answer: c

Explanation: A pure virtual function does not have a definition corresponding to base class. All derived class may or may not have an implementation of a pure virtual function. there is no pure keyword in C++.

5. Pick the correct statement.

- a) Pure virtual functions and virtual functions are the same
- b) Both Pure virtual function and virtual function have an implementation in the base class
- c) Pure virtual function has no implementation in the base class whereas virtual function may have an implementation in the base class
- d) The base class has no pure virtual function

View Answer

Answer: c

Explanation: Pure virtual function has no implementation in the base class whereas virtual function may have an implementation in the base class. The base class has at least one pure virtual function.

- a) Class B
- b) Error
- c) Segmentation fault
- d) No output

View Answer

Answer: a

Explanation: The program is correct so no error occurs hence the program runs successfully and b is calling is func() function therefore "Class B" is printed.

- a) Class B
- b) Error
- c) Segmentation fault
- d) No output

View Answer

Answer: b

Explanation: The C++ does allows to declare a normal object for an abstract class therefore the program throws an error as we are trying to declare an object of abstract class.

- a) Class B
- b) Error
- c) Segmentation fault
- d) No output

View Answer

Answer: c

Explanation: As we are allowed to declare a pointer object of an abstract so the program does not give any compilation error but as there is no definition of func() function corresponding to class A, therefore, the program gives segmentation fault as it is not able to call such function from class A.

C++ Programming Questions and Answers – Design of Class Hierarchies

1. Which interface determines how your class will be used by another program?

- a) public
- b) private
- c) protected
- d) none of the mentioned

View Answer

Answer: a

Explanation: If we invoked the interface as public means, We can access the program from other programs also.

2. Pick out the correct statement about the override.

- a) Overriding refers to a derived class function that has the same name and signature as a base class virtual function
- b) Overriding has different names
- c) Overriding refers to a derived class
- d) None of the mentioned

View Answer

Answer: a

Explanation: None.

3. How many ways of reusing are there in the class hierarchy?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: b

Explanation: Class hierarchies promote reuse in two ways. They are code sharing and interface sharing.

4. How many types of class are there in c++?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: c

Explanation: There are three types of classes. They are abstract base classes, concrete derived classes, standalone classes.

- a) 10
- b) 4
- c) 40
- d) None of the mentioned

View Answer

Answer: c

Explanation: In this program, We are multiplying the value 10 and 4 by using inheritance.

Output:

```
$ g++ des.cpp
$ a.out
40
```

6. Pick out the correct statement about multiple inheritances.

- a) Deriving a class from one direct base class
- b) Deriving a class from more than one direct base class
- c) Deriving a class from more than one direct derived class
- d) None of the mentioned

View Answer

Answer: b

Explanation: In multiple inheritances, We are able to derive a class from more than one base class.

- a) 10
- b) 20
- c) 1020
- d) None of the mentioned

View Answer

Answer: c

Explanation: In this program, We are passing the values from the main class and printing it on the inherited classes.

Output:

advertisement

```
$ g++ des2.cpp
$ a.out
1020
```

- a) 123
- b) 12
- c) 213
- d) 321

View Answer

Answer: a

Explanation: We are passing the objects and executing them in a certain order and we are printing the program flow.

Output:

```
$ g++ des3.cpp
$ a.out
123
```

9. What does inheritance allow you to do?

- a) create a class
- b) create a hierarchy of classes
- c) access methods
- d) none of the mentioned

View Answer

Answer: b

Explanation: Inheritance helps in creating hierarchy of classes by making connections between different classes in which one is called base class and other is class derived class.

10. What is the syntax of inheritance of class?

- a) class name
- b) class name: access specifier
- c) class name: access specifier class name
- d) none of the mentioned

View Answer

Answer: c

Explanation: Syntax is:

class Class_Name: Access_Specifier Base_Class_Name

example:

class A {};

class B: public A {};

ayzom.com

C++ Programming Questions and Answers – Class Hierarchies and Abstract Classes

1. How many kinds of classes are there in c++?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: b

Explanation: There are two kinds of classes in c++. They are absolute class and the concrete class.

2. What is meant by polymorphism?

- a) class having many forms
- b) class having only single form
- c) class having two forms
- d) none of the mentioned

View Answer

Answer: a

Explanation: Polymorphism is literally meant class having many forms.

3. How many types of inheritance are there in c++?

- a) 2
- b) 3
- c) 4
- d) 5

View Answer

Answer: d

Explanation: There are five types of inheritance in c++. They are single, Multiple, Hierarchical, Multilevel, Hybrid.

a) 66.5

b) 64.5

c) 62.5

d) 60.5

View Answer

Answer: a

Explanation: In this program, We are passing the values by using different methods and totaling the marks to get the result.

Output:

```
$ g++ class.cpp
$ a.out
Total Score=66.5
```

a) 1212

b) 1210

c) 1010

d) none of the mentioned

View Answer

Answer: b

Explanation: In this program, We are calculating the area of rectangle and triangle by using multilevel inheritance.

```
$ g++ class1.cpp  
$ a.out  
1210
```

6. What is meant by container ship?

- a) class contains objects of other class types as its members
- b) class contains objects of other class types as its objects
- c) class contains objects of other class types as its members & also objects
- d) none of the mentioned

View Answer

Answer: a

Explanation: None.

7. How many types of the constructor are there in C++?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: c

Explanation: There are three types of constructor in C++. They are the Default constructor, Parameterized constructor, Copy constructor.

8. How many constructors can present in a class?

- a) 1
- b) 2
- c) 3
- d) multiple

View Answer

Answer: d

Explanation: There can be multiple constructors of the same class, provided they have different signatures.

9. What should be the name of the constructor?

- a) same as the object
- b) same as the member
- c) same as the class
- d) none of the mentioned

View Answer

Answer: c

Explanation: None.

10. What does derived class does not inherit from the base class?

- a) constructor and destructor
- b) friends
- c) operator = () members
- d) all of the mentioned

View Answer

Answer: d

Explanation: The derived class inherits everything from the base class except the given things.

C++ Programming Questions and Answers – Simple String Template

1. What is a template?

- a) A template is a formula for creating a generic class
- b) A template is used to manipulate the class
- c) A template is used for creating the attributes
- d) None of the mentioned

View Answer

Answer: a

Explanation: Templates are used to for creating generic classes to handle different types in single classes.

2. Pick out the correct statement about string template.

- a) It is used to replace a string
- b) It is used to replace a string with another string at runtime
- c) It is used to delete a string
- d) None of the mentioned

View Answer

Answer: b

Explanation: Every string template is used to replace the string with another string at runtime.

3. How to declare a template?

- a) tem
- b) temp
- c) template<>
- d) none of the mentioned

View Answer

Answer: c

Explanation: template<> syntax is used.

An example for calculating max of two ints, floats, doubles, or any other number type where T indicates the type of the parameters passes.

template <typename T>

```
T max(T a, T b){  
    return a > b? a : b;  
}
```

- a) 416AA
- b) 164AA
- c) AA416
- d) none of the mentioned

View Answer

Answer: a

Explanation: In this program, We are using two template to calculate the square and to find the addition.

Output:

```
$ g++ tem.cpp  
$ a.out  
416AA
```

- a) 23
- 2.1 4.41
- b) 24
- 2.1 4.41
- c) 24

2.1 3.41

d) none of the mentioned

View Answer

Answer: b

Explanation: In this multiple templated types, We are passing two values of different types and producing the result.

Output:

```
$ g++ tem1.cpp
$ a.out
24
2.1 4.41
```

a) 5.5

Hello World

b) 5.5

c) Hello World

d) None of the mentioned

View Answer

Answer: a

Explanation: In this program, We are passing the value to the template and printing it in the template.

Output:

```
$ g++ tem2.cpp
$ a.out
5.5
Hello World
```

7. How many types of templates are there in c++?

a) 1

b) 2

c) 3

d) 4

View Answer

Answer: b

Explanation: There are two types of templates. They are function template and class template.

8. Which are done by compiler for templates?

a) type-safe

b) portability

c) code elimination

d) all of the mentioned

View Answer

Answer: a

Explanation: The compiler can determine at compile time whether the type associated with a template definition can perform all of the functions required by that template definition.

9. What may be the name of the parameter that the template should take?

a) same as template

b) same as class

c) same as function

d) none of the mentioned

View Answer

Answer: a

Explanation: None.

10. How many parameters are legal for non-type template?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: d

Explanation: The following are legal for non-type template parameters: integral or enumeration type, Pointer to object or pointer to function, Reference to object or reference to function, Pointer to member.

ayzom.com

C++ Programming Questions and Answers – Function Templates

1. What is a function template?

- a) creating a function without having to specify the exact type
- b) creating a function with having a exact type
- c) all of the mentioned
- d) none of the mentioned

View Answer

Answer: a

Explanation: None.

2. Which is used to describe the function using placeholder types?

- a) template parameters
- b) template type parameters
- c) template type
- d) none of the mentioned

View Answer

Answer: b

Explanation: During runtime, We can choose the appropriate type for the function and it is called as template type parameters.

3. Pick out the correct statement.

- a) you only need to write one function, and it will work with many different types
- b) it will take a long time to execute
- c) duplicate code is increased
- d) none of the mentioned

View Answer

Answer: a

Explanation: Because of template type parameters, It will work with many types and saves a lot of time.

- a) 100
- b) 200
- c) 300
- d) 100200

View Answer

Answer: b

Explanation: In this program, We are returning the maximum value by using function template.

Output:

```
$ g++ ftemp.cpp
$ a.out
200
```

- a) 200
- 3.123
- b) 3.123
- 200
- c) 200
- d) 3.123

View Answer

Answer: a

Explanation: In this program, We are passing the values and getting it back from template. And we are using the constructor and destructor for the function template.

Output:

```
$ g++ ftempl.cpp
$ a.out
200
3.123
```

- a) 100
- b) 200
- c) 50
- d) none of the mentioned

View Answer

Answer: b

Explanation: In this program, We are using class to pass the value and then we are manipulating it.

Output:

```
$ g++ ftemp3.cpp
$ a.out
200
```

- a) 2 4
- 2.2 4.84
- b) 2 4
- c) error
- d) runtime error

View Answer

Answer: a

Explanation: In this program, We are passing the values and calculating the square of the value by using the function template.

Output:

advertisement

```
$ g++ ftemp4.cpp
$ a.out
2 4
2.2 4.84
```

- a) 2.1
- b) 3.1
- c) 3.2
- d) 2.1
- 3.1
- 4.1

View Answer

Answer: d

Explanation: In this program, We are using the for loop to increment the value by 1 in the function template.

Output:

```
$ g++ ftemp5.cpp
$ a.out
2.1
3.1
4.1
```

9. What can be passed by non-type template parameters during compile time?

- a) int
- b) float
- c) constant expression
- d) none of the mentioned

View Answer

Answer: c

Explanation: Non-type template parameters provide the ability to pass a constant expression at compile time. The constant expression may also be an address of a function, object or static class member.

10. From where does the template class derived?

- a) regular non-templated C++ class
- b) templated class
- c) regular non-templated C++ class or templated class
- d) none of the mentioned

View Answer

Answer: c

Explanation: None.

ayzom.com

C++ Programming Questions and Answers – Function Templates – 2

1. What are Templates in C++?

- a) A feature that allows the programmer to write generic programs
- b) A feature that allows the programmer to write specific codes for a problem
- c) A feature that allows the programmer to make program modular
- d) A feature that does not add any power to the language

View Answer

Answer: a

Explanation: Templates are features in C++ that allows the programmer to write generic programs. for example, making the same function to take different types of arguments and perform the same action on them without specifying the type in the argument list.

2. In how many ways templates concept can be used?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: b

Explanation: Template concept can be used in two different ways. They are function templates used with functions and class templates used with classes.

3. What is the difference between normal function and template function?

- a) The normal function works with any data types whereas template function works with specific types only
- b) Template function works with any data types whereas normal function works with specific types only
- c) Unlike a normal function, the template function accepts a single parameter
- d) Unlike the template function, the normal function accepts more than one parameters

View Answer

Answer: b

Explanation: As a template feature allows you to write generic programs. therefore a template function works with any type of data whereas normal function works with the specific types mentioned while writing a program. Both normal and template function accepts any number of parameters.

4. Templates simulate which of the following feature?

- a) Polymorphism
- b) Abstraction
- c) Encapsulation
- d) Inheritance

View Answer

Answer: a

Explanation: Template function helps in writing functions that work with different types of parameters which is what polymorphism means i.e. using same function prototype to perform the same operations on different types of parameters.

5. Which keyword is used for the template?

- a) Template
- b) template
- c) Temp
- d) temp

View Answer

Answer: b

Explanation: C++ uses template reserved keyword for defining templates.

6. What is the correct syntax of defining function template/template functions?

- a) `template <class T> void(T a){cout<<a;}`
- b) `Template <class T> void(T a){cout<<a;}`
- c) `template <T> void(T a){cout<<a;}`
- d) `Template <T> void(T a){cout<<a;}`

View Answer

Answer: a

Explanation: Starts with keyword template and then <class VAR>, then use VAR as type anywhere in the function below.

- a) A function taking a single generic parameter and returning a generic type
- b) A function taking a single generic parameter and returning nothing
- c) A function taking single int parameter and returning a generic type
- d) A function taking a single generic parameter and returning a specific non-void type

View Answer

Answer: b

Explanation: As the return type of function is void therefore function is not returning anything. Now as the function is taking a template T as its argument which is a general type, therefore, it is accepting a single general type argument.

- a) A function taking a single generic parameter and returning a generic type
- b) A function taking a single generic parameter and returning nothing
- c) A function taking single int parameter and returning a generic type
- d) A function taking a single generic parameter and returning a specific non-void type

View Answer

Answer: a

Explanation: As the return type of function is template T, therefore, the function is returning a general type. Now as the function is taking a template T as its argument which is a general type, therefore, it is accepting a single general type argument.

- a) A function taking a single generic parameter and returning a generic type which may be different from argument type
- b) A function taking a single generic parameter and returning a generic type which must be different from argument type
- c) A function taking a single generic parameter and returning a generic type which must have the same type as argument type
- d) A function taking a single generic parameter and returning a specific non-void type

View Answer

Answer: a

Explanation: As the return type of function is template U, therefore, the function is returning a general type. Now as the function is taking a template T as its argument which is a general type, therefore, it is accepting a single general type argument. But as U and T are different therefore return type and argument type may be the same or different. Same if U and T both have the same type and different if both have different types.

- a) 5
- b) 55
- c) Error
- d) Segmentation fault

View Answer

Answer: c

Explanation: Template function cannot be overloaded as done in this program. They can be overloaded with normal functions or other templates.

- a) 5
- b) 55.5

- c) Error
 - d) Segmentation fault
- View Answer

Answer: b

Explanation: C++ allows to overload template functions with normal functions therefore the program does not gives any error. When program is executed then `int b = func(a);` calls the normal function whereas `float c = func(5.5)` calls the template function.

- a) 5
 - b) 55
 - c) Error
 - d) Segmentation fault
- View Answer

Answer: b

Explanation: C++ allows such overlaoding of template functions therefore the program does not gives any error. The first template function returns the same parameter, whereas second template function returns type casted value.

View Answer

Answer: a

Explanation: We need two templates one for handling numbers and other for operators. Here T handles numbers and U handles operators. T can take any number type, whereas U can take either char or bool type. The function takes numbers and operators and produces the result. The return type should be a number, therefore, the return type is T.

- a) replace templates T, U with auto keyword
 - b) replace templates T, U with generic keyword
 - c) replace templates T, U with temp keyword
 - d) replace templates T, U with GEN_TEMP keyword
- View Answer

Answer: a

Explanation: C++ allows use of auto keyword which helps in binding types dynamically. The modified without template version of above program is given below:

```
=====
auto func(auto a, auto b, auto c)
{
    if(c == '+' || c){
        return a+b;
    }
    else if(c == '-' || !c){
        return a-b;
    }
}
=====
```

- a) `func(3,'a');`
 - b) `func(3,'a');`
 - c) `func(3,'a');`
 - d) `func(3,'a');`
- View Answer

Answer: a

Explanation: This is the correct method of explicit template function call.

C++ Programming Questions and Answers – Class Templates

1. What is the syntax of class template?

- a) template <paramaters> class declaration
- b) Template <paramaters> class declaration
- c) temp <paramaters> class declaration
- d) Temp <paramaters> class declaration

View Answer

Answer: a

Explanation: Syntax involves template keyword followed by list of parameters in angular brackets and then class declaration. As follows
template <paramaters> class declaration;

- c) Compile-time error
- d) Run-time error

View Answer

Answer: c

Explanation: As class A is a template class and a template class during object declaration requires template arguments, therefore A a; gives an error.

- c) Compile-time error
- d) Run-time error

View Answer

Answer: a

Explanation: In this program object of template class has an argument, therefore, the program does not give any error. Hence the program runs perfectly and the output is produced.

Output:

```
$ ./a.out
Created
Destroyed
```

View Answer

Answer: b

Explanation: All the objects are first created and then all are destroyed when the scope of all the objects are destroyed i.e. at the end of the main function. That's why first all the created are printed and then after that, all the destroyed are printed.

View Answer

Answer: a

Explanation: As a1 object is defined with as the template parameter therefore all the numbers passed to the func() are converted to integers and as integer division of 3 by 2 is 1 therefore the output is 1.

View Answer

Answer: d

Explanation: As a1 object is defined with as the template parameter therefore all the numbers passed to the func() are converted to decimal floating point numbers and as floating division of 3 by 2 is 1.5 therefore the output is 1.5 for both 1st and 2nd call of the function.

View Answer

Answer: a

Explanation: As a1 object is defined with as the template parameter therefore all the numbers passed to the func() are converted to characters integers. So when 1st func() call is done with 65 and 1 then after dividing 65 by 1 gives 65 for which char equivalent is 'A' hence 'A' is printed similarly in 2nd case $65.28/1.1 = 65$ integer which represents 'A' in char type therefore 'A' is printed.

View Answer

Answer: a

Explanation: class A has a variable of type template type therefore the type of the class depends on the template type assigned. Therefore for char template the class size is 1. For int template type size is 4 and for double type size is 8.

9. How the template class is different from the normal class?

- a) Template class generate objects of classes based on the template type
- b) Template class saves system memory
- c) Template class helps in making generic classes
- d) All of the mentioned

View Answer

Answer: d

Explanation: Size of the object of template class varies depending on the type of template parameter passed to the class. Due to which each object occupies different memories on system hence saving extra memories. Template class also helps in making generic classes.

- b) nothing
- c) Error
- d) Segmentation fault

View Answer

Answer: b

Explanation: The program is correct therefore no error or segmentation fault but as no print() function is called using any object therefore nothing is printed on the output console.

- b) Nothing
- c) Error
- d) Segmentation fault

View Answer

Answer: a

Explanation: As we have defined the default templates so when a1 is initialized with 5 and 10 as we have specified <int,int> as the parameters. For a2 5 and \$ is initialized as we have only passed <int> as the template parameter. Similar goes for a3. Hence the output is printed as mentioned.

```
$ ./a.out
5 10
5 $
10 $
```

12. How many template parameters are allowed in template classes?

- a) 1
- b) 2
- c) 3
- d) one or more

View Answer

Answer: d

Explanation: Just like normal parameters we can pass more than one or more template parameters to a template class.

C++ Programming Questions and Answers – Template Arguments to Specify Policy Usage

1. What is meant by template parameter?

- a) It can be used to pass a type as argument
- b) It can be used to evaluate a type
- c) It can of no return type
- d) None of the mentioned

View Answer

Answer: a

Explanation: A template parameter is a special kind of parameter that can be used to pass a type as argument.

2. Which keyword can be used in template?

- a) class
- b) typename
- c) both class & typename
- d) function

View Answer

Answer: c

Explanation: Both keywords can be used as shown below:

template <class T> function declaration;

template <typename T> function declaration;

3. What is the validity of template parameters?

- a) inside that block only
- b) inside the class
- c) whole program
- d) any of the mentioned

View Answer

Answer: a

Explanation: Template parameters are valid inside a block only i.e. they have block scope.

- a) 100
- b) 3.1416
- c) 100
3.1416
- d) none of the mentioned

View Answer

Answer: c

Explanation: In this program, We are printing the integer in the first function and float in the second function.

Output:

```
$ g++ farg.cpp
$ a.out
100
3.1416
```

- a) 6
- b) 6
10
- c) 5
10
- d) 6

5

View Answer

Answer: b

Explanation: In this program, We are using the ternary operator on the template function.

Output:

```
$ g++ farg.cpp
$ a.out
6
10
```

a) 100

b) 200

c) 3.123

d) 2003.123

View Answer

Answer: d

Explanation: In this program, We are passing the value and returning it from template.

Output:

```
$ g++ farg3.cpp
$ a.out
2003.123
```

a) 2.1

b) 3.1

c) 4.1

d) 2.1

3.1

4.1

View Answer

Answer: d

Explanation: In this program, We are using the non-type template parameter to increment the value in the function template.

Output:

```
$ g++ farg4.cpp
$ a.out
2.1
3.1
4.1
```

8. Why we use :: template-template parameter?

a) binding

b) rebinding

c) both binding & rebinding

d) none of the mentioned

View Answer

Answer: c

Explanation: It is used to adapt a policy into binary ones.

9. Which parameter is legal for non-type template?

a) pointer to member

b) object

c) class

d) none of the mentioned

View Answer

Answer: a

Explanation: The following are legal for non-type template parameters: integral or enumeration type, Pointer to object or pointer to function, Reference to object or reference to function, Pointer to member.

10. Which of the things does not require instantiation?

- a) functions
- b) non virtual member function
- c) member class
- d) all of the mentioned

View Answer

Answer: d

Explanation: The compiler does not generate definitions for functions, non virtual member functions, class or member class because it does not require instantiation.

ayzom.com

C++ Programming Questions and Answers – Specialization

1. What is meant by template specialization?

- a) It will have certain data types to be fixed
- b) It will make certain data types to be dynamic
- c) Certain data types are invalid
- d) None of the mentioned

View Answer

Answer: a

Explanation: In the template specialization, it will make the template to be specific for some data types.

2. Which is similar to template specialization?

- a) template
- b) function overloading
- c) function template overloading
- d) none of the mentioned

View Answer

Answer: c

Explanation: None

3. Which is called on allocating the memory for the array of objects?

- a) destructor
- b) constructor
- c) method
- d) none of the mentioned

View Answer

Answer: b

Explanation: When you allocate memory for an array of objects, the default constructor must be called to construct each object. If no default constructor exists, you're stuck needing a list of pointers to objects.

a) 2: 4AA:

b) 2:4

c) AA

d) 2:4A

View Answer

Answer: a

Explanation: Template specialization is used when a different and specific implementation is to be used for a specific data type. In this program, We are using integer and character.

Output:

```
$ g++ spec.cpp
$ a.out
2: 4AA:
```

a) 2.1

b) 378.228

c) 2.1: 378.228

d) None of the mentioned

View Answer

Answer: c

Explanation: In this program, We specify the type in the template function. We need to compile this program by adding -std=c++0x.

Output:

```
$ g++ -std=c++0x spec1.cpp
$ a.out
2.1: 378.228
```

- a) 1
- b) 1.2
- c) 1
- 1.2
- d) 1
- 1
- 1.2

View Answer

Answer: d

Explanation: In this program, We are passing the value of specified type and printing it by specialization.

Output:

```
$ g++ spec2.cpp
$ a.out
1
1
1.2
```

- a) template
- b) class
- c) no specialization
- d) none of the mentioned

View Answer

Answer: a

Explanation: In this program, We are computing the result in the specialized block of the program.

Output:

```
$ g++ spec3.cpp
$ a.out
template
```

- a) 6
- b) 10
- c) 6default10
- d) None of the mentioned

View Answer

Answer: c

Explanation: In this program, We are defining three templates and specializing it and passing the values to it and printing it.

Output:

```
$ g++ spec5.cpp
$ a.out
6default10
```

9. How many types of specialization are there in c++?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: b

Explanation: There are two types of specialization. They are full specialization and partial specialization.

10. What is another name of full specialization?

- a) explicit specialization
- b) implicit specialization
- c) function overloading template
- d) none of the mentioned

View Answer

Answer: a

Explanation: None

ayzom.com

C++ Programming Questions and Answers – Derivation and Templates

1. Which is dependant on template parameter?

- a) base class
- b) abstract class
- c) method
- d) none of the mentioned

View Answer

Answer: a

Explanation: None

2. Which value is placed in the base class?

- a) derived values
- b) default type values
- c) both default type & derived values
- d) none of the mentioned

View Answer

Answer: b

Explanation: We can place the default type values in a base class and overriding some of them through derivation.

3. How many bits of memory needed for internal representation of class?

- a) 1
- b) 2
- c) 4
- d) no memory needed

View Answer

Answer: d

Explanation: classes that contain only type members, nonvirtual function members, and static data members do not require memory at run time.

- a) True
- b) error
- c) no output
- d) runtime error

View Answer

Answer: a

Explanation: In this program, We are passing the values and inheriting it to the other class and printing the result.

```
$ g++ dert.cpp
$ a.out
True
```

- a) 42
- b) A
- c) 42
A
- d) A
42

View Answer

Answer: c

Explanation: In this program, We are passing the values by using the template inheritance and printing it.

Output:

```
$ g++ dert.cpp
$ a.out
42
A
```

- a) 100
 - b) 200
 - c) error
 - d) runtime error
- View Answer

Answer: b

Explanation: In this program, We are passing the values and manipulating it by using the template inheritance.

Output:

```
$ g++ dert2.cpp
$ a.out
200
```

- a) k
 - b) l
 - c) error
 - d) runtime error
- View Answer

Answer: a

Explanation: In this program, We are passing the values and printing it by using template inheritance.

Output:

```
$ g++ dert3.cpp
$ a.out
k
```

- a) 1234
 - b) 4321
 - c) 1423
 - d) 1342
- View Answer

Answer: d

Explanation: In this program, We are printing the order of execution of constructor and destructor in the class.

Output:

```
$ g++ dert4.cpp
$ a.out
1342
```

9. How many kinds of entities are directly parameterized in c++?

- a) 1
 - b) 2
 - c) 3
 - d) 4
- View Answer

Answer: c

Explanation: C++ allows us to parameterize directly three kinds of entities through templates: types, constants, and templates.

10. How many kinds of parameters are there in C++?

- a) 1
- b) 2
- c) 3
- d) None of the mentioned

[View Answer](#)

Answer: c

Explanation: There are three kinds of parameters are there in C++. They are type, non-type, template.

ayzom.com

C++ Programming Questions and Answers – Standard Template Library

1. What is the Standard Template Library?

- a) Set of C++ template classes to provide common programming data structures and functions
- b) Set of C++ classes
- c) Set of Template functions used for easy data structures implementation
- d) Set of Template data structures only

View Answer

Answer: a

Explanation: STL expanded as Standard Template Library is set of C++ template classes to provide common programming data structures and functions.

2. Pick the correct statement.

- a) STL is a generalized library
- b) Components of STL are parameterized
- c) STL uses the concept of templates classes and functions to achieve generalized implementation
- d) All of the mentioned

View Answer

Answer: d

Explanation: STL is a generalized library and components of STL are parameterized. STL uses the concept of templates classes and function to achieve generalized implementation.

3. How many components STL has?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: d

Explanation: STL has four components namely Algorithms, Containers, Functors and Iterators.

4. What are the containers?

- a) Containers store objects and data
- b) Containers stores all the algorithms
- c) Containers contain overloaded functions
- d) Containers contain set of Iterators

View Answer

Answer: a

Explanation: Containers is a component of STL which stores objects and data.

5. In how many categories, containers are divided?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: d

Explanation: Containers are divided into 4 categories namely Sequence Containers, Associative Containers, Unordered Associative Containers and Container Adaptors.

6. What are the Sequence Containers?

- a) Containers that implements data structures which can be accessed sequentially
- b) Containers that implements sorted data structures for fast search in $O(\log n)$
- c) Containers that implements unsorted(hashd) data structures for quick search in $O(1)$
- d) Containers that implements data structures which can be accessed non-sequentially

View Answer

Answer: a

Explanation: Sequence Containers is the subset of Containers that implements data structures which can be accessed sequentially.

7. How many Sequence Containers are provided by C++?

- a) 2
- b) 3
- c) 4
- d) 5

View Answer

Answer: d

Explanation: C++ provides 5 types of Sequence Containers namely array, vector, deque, forward_list and list.

8. What are the Associative Containers?

- a) Containers that implements data structures which can be accessed sequentially
- b) Containers that implements sorted data structures for fast search in $O(\log n)$
- c) Containers that implements unsorted(hashd) data structures for quick search in $O(1)$
- d) Containers that implements data structures which can be accessed non-sequentially

View Answer

Answer: b

Explanation: Associative Containers is the subset of Containers that implements sorted data structures for fast search in $O(\log n)$.

9. How many Associative Containers are provided by C++?

- a) 2
- b) 3
- c) 4
- d) 5

View Answer

Answer: c

Explanation: C++ provides 4 types of Associative Containers namely Set, Map, multiset and multimap.

10. What are Unordered Associative Containers?

- a) Containers that implements data structures which can be accessed sequentially
- b) Containers that implements sorted data structures for fast search in $O(\log n)$
- c) Containers that implements unsorted(hashd) data structures for quick search in $O(1)$
- d) Containers that implements data structures which can be accessed non-sequentially

View Answer

Answer: c

Explanation: Unordered Associative Containers is the subset of Containers that implements unsorted(hashd) data structures for quick search in $O(1)$ amortized $O(n)$ Worst case complexity.

11. What are Container Adaptors?

- a) Containers that implements data structures which can be accessed sequentially
- b) Containers that implements sorted data structures for fast search in $O(\log n)$
- c) Containers that implements unsorted(hashd) data structures for quick search in $O(1)$
- d) Containers that provide a different interface for sequential containers

View Answer

Answer: d

Explanation: Container Adaptors is the subset of Containers that provides a different interface for sequential containers.

12. How many Container Adaptors are provided by C++?

- a) 2
- b) 3
- c) 4
- d) 5

View Answer

Answer: b

Explanation: C++ provides 3 types of Container Adaptors namely Stack, Queue and Priority Queues.

13. What are Iterators?

- a) Iterators are used to iterate over C-like arrays
- b) Iterators are used to iterate over pointers
- c) Iterators are used to point memory addresses of STL containers
- d) Iterators are used to iterate over functions

View Answer

Answer: c

Explanation: In C++, Iterators are provided to iterate over the STL containers.

14. How many types of Iterators are provided by C++?

- a) 2
- b) 3
- c) 4
- d) 5

View Answer

Answer: d

Explanation: There are five types of Iterators provided by C++ namely Input Iterators, Output Iterators, Forward Iterators, Bi-directional Iterators and Random-access Iterators.

15. Which header file is used for Iterators?

- a) <iterator>
- b) <algorithm>
- c) <iter>
- d) <loopIter>

View Answer

Answer: a

Explanation: Iterators are present inside the <iterator> header file so this header file is needed to use Iterators.

C++ Programming Questions and Answers – Templates

1. Which of the following is used for generic programming?

- a) Virtual functions
- b) Modules
- c) Templates
- d) Abstract Classes

View Answer

Answer: c

Explanation: Templates are used for generic programming. They help in making generic functions and classes hence achieving the generic codes.

2. Which of the following is correct about templates?

- a) It is a type of compile time polymorphism
- b) It allows the programmer to write one code for all data types
- c) Helps in generic programming
- d) All of the mentioned

View Answer

Answer: d

Explanation: Templates are used for generic programming hence allowing to write a single function for all data types. It is a type of compile time polymorphism.

View Answer

Answer: a

Explanation: As template is a type of polymorphism so count becomes 1 for int type because we have called the function for int twice but as we have called it only once for double therefore value of count is 0 for double i.e. for last call.

d) Error

View Answer

Answer: d

Explanation: Here in the template function as both the argument have same type T but in third called of max function we are using 2 types int and float while calling the function hence the program gives error.

View Answer

Answer: a

Explanation: In this as we have first used int to define the objects of class Test so count is increased to 2 for int types and declared the double object once therefore the constructor is called only once hence value of count is 1.

View Answer

Answer: a

Explanation: In this we have made class with two template arguments which are used in defining one variable of each type inside the class. So the total size of the class will depend on the type of U and T. So when we have defined the object with <char,char> type then the class have two variable of char hence making the size of class = 2 which in case of <int,int> = 8.

c) Error

d) Segmentation fault

View Answer

Answer: a

Explanation: In this case the class has three variables of type T, U and V = double by default. Now as we have defined objects with types <int,int> so its size is 4+4+8 = 16 and second object of <double,double> which leads to 8+8+8 = 24.

d) Error

View Answer

Answer: a

Explanation: In this, we are passing arr and its size in the template function. Now during first call value of m becomes 10 as it is the minimum in the array. In second call the value of m becomes 1 which is minimum in 2nd array.

- a) Template Called 20
 - b) Called 20
 - c) Error
 - d) Segmentation fault
- View Answer

Answer: b

Explanation: For T = int we have created a separate definition for the above template function. Hence the call using int takes the newly defined function.

- a) 1
 - b) 1024
 - c) Error
 - d) Segmentation fault
- View Answer

Answer: b

Explanation: The above call for struct will call the first struct for $n > 0$ and second one when $n = 0$. Therefore when value of $n = 10$ the until n becomes 0 first struct is called so we will call $2 * 2 * \dots 10 \text{ times} * 1$ which will give the result $2^{10} = 1024$.

ayzom.com

C++ Programming Questions and Answers – Error Handling

1. Which keyword is used to handle the exception?

- a) try
- b) throw
- c) catch
- d) none of the mentioned

View Answer

Answer: c

Explanation: When we found an exception in the program, we need to throw that and we handle that by using the catch keyword.

2. Which is used to throw an exception?

- a) throw
- b) try
- c) catch
- d) none of the mentioned

View Answer

Answer: a

Explanation: throw keyword is used to throw an exception.

eg:

```
if(divisor == 0){  
    throw "Divide by zero error";  
}
```

3. What is the use of the 'finally' keyword?

- a) It is used to execute at the starting of the program
- b) It will be executed at the end of the program even if the exception arises
- c) It will be executed at the starting of the program even if the exception arises
- d) none of the mentioned

View Answer

Answer: b

Explanation: finally keyword will be executed at the end of all the exceptions.

- a) 50
- b) 0
- c) Division by zero condition!
- d) Error

View Answer

Answer: c

Explanation: It's a mathematical certainty, we can't divide by zero, so we're arising an exception.

Output:

```
$ g++ excep.cpp  
$ a.out  
Division by zero condition!
```

- a) 20
- b) An exception occurred
- c) Error
- d) An exception occurred 20

View Answer

Answer: d

Explanation: We are handling the exception by throwing that number. So the output is printed with the given number.

Output:

```
$ g++ excep1.cpp
$ a.out
An exception occurred 20
```

- a) Exception
 - b) Error
 - c) My exception
 - d) runtime error
- View Answer

Answer: c

Explanation: This is a standard exception handler used in the class.

Output:

```
$ g++ excep2.cpp
$ a.out
My exception
```

- a) Allocated
 - b) Standard exception
 - c) Depends on the memory
 - d) Error
- View Answer

Answer: c

Explanation: In this program, We are allocating the memory for array. If it is allocated means, no exception will arise and if there is no size in memory means, Exception will arise.

Output:

advertisement

```
$ g++ excep3.cpp
$ a.out
allocated
```

- a) C++ Exception
 - b) Exception caught
 - c) Exception caught
- C++ Exception
- d) Error
- View Answer

Answer: c

Explanation: We are defining the user-defined exception in this program.

Output:

```
$ g++ excep4.cpp
$ a.out
C++ Exception
Exception caught
```

9. How do define the user-defined exceptions?

- a) inheriting and overriding exception class functionality
- b) overriding class functionality
- c) inheriting class functionality

d) none of the mentioned

View Answer

Answer: a

Explanation: User defined exceptions can be done by inheriting and overriding the exception class functionality.

10. Which exception is thrown by `dynamic_cast`?

a) `bad_cast`

b) `bad_typeid`

c) `bad_exception`

d) `bad_alloc`

View Answer

Answer: a

Explanation: `bad_cast` exception is thrown by `dynamic_cast`.

ayzom.com

C++ Programming Questions and Answers – Grouping of Exceptions

1. How many types of exception handling are there in c++?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: b

Explanation: There are two types of exception handling in c++. They are synchronous exception handling and asynchronous exception handling.

2. How many runtime error messages associated with exception?

- a) 2
- b) 4
- c) 5
- d) infinite

View Answer

Answer: b

Explanation: There are four runtime error messages associated with exceptions. They are `overflow_error`, `range_error`, `system_error` and `underflow_error`.

3. Which block should be placed after try block?

- a) catch
- b) throw
- c) either catch or throw
- d) none of the mentioned

View Answer

Answer: a

Explanation: Syntax of try catch block:

```
try{
// do something
}
catch{
// catch respective error.
}
finally{
// do something after trying or catching error i.e. run this block in both cases.
}
```

- a) 10
- b) 2
- c) Bad Operator
- d) $10 / 5 = 2$

View Answer

Answer: d

Explanation: In this program, We are dividing the two variables and printing the result. If any one of the operator is zero means, it will arise a exception.

Output:

```
$ g++ gex.cpp
$ a.out
10 / 5 =2
```

- a) No exception
- b) exception number
- c) exception number: 1
- d) none of the mentioned

View Answer

Answer: c

Explanation: If we caught a integer value means, there will be an exception, if it is not a integer, there will not be a exception.

Output:

```
$ g++ gex1.cpp
$ a.out
exception number: 1
```

- a) 10
- b) -3
- c) 15
- d) none of the mentioned

View Answer

Answer: b

Explanation: We are manipulating the values, if there is any infinite value means, it will raise an exception.

Output:

```
$ g++ gex2.cpp
$ a.out
-3
```

- a) integer:10character:x
- b) integer:10
- c) character:x
- d) none of the mentioned

View Answer

Answer: a

Explanation: We are passing the integer and character and catching it by using multiple catch statement.

Output:

advertisement

```
$ g++ gex3.cpp
$ a.out
integer:10character:x
```

- a) 1
- b) exception: 2
- c) 1
- exception: 2
- d) none of the mentioned

View Answer

Answer: c

Explanation: In this program, We are printing one and raising a exception at 2.

Output:

```
$ g++ gex4.cpp
$ a.out
1
exception: 2
```

9. Pick out the correct Answer.

- a) Exceptions are not suitable for critical points in code
- b) Exception are suitable for critical points in code
- c) All of the mentioned
- d) None of the mentioned

View Answer

Answer: a

Explanation: If there is many number of exceptions in the program means, We have to use multiple catch statement and it is hard to keep track of the program.

10. When exceptions are used?

- a) To preserve the program
- b) Exceptions are used when postconditions of a function cannot be satisfied
- c) Exceptions are used when postconditions of a function can be satisfied
- d) None of the mentioned

View Answer

Answer: c

Explanation: None.

ayzom.com

C++ Programming Questions and Answers – Catching Exceptions

1. How many parameters does the throw expression can have?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: a

Explanation: In c++ program, We can be able to throw only one error at a time.

2. Where exception are handled?

- a) inside the program
- b) outside the regular code
- c) both inside or outside
- d) none of the mentioned

View Answer

Answer: b

Explanation: none

3. Which is used to check the error in the block?

- a) try
- b) throw
- c) catch
- d) none of the mentioned

View Answer

Answer: a

Explanation: The try block is used to check for errors, if there is any error means, it can throw it to catch block.

- a) exception arised
- b) error
- c) exception
- d) runtime error

View Answer

Answer: a

Explanation: In this program, We are arising a standard exception and catching that and returning a statement.

Output:

```
$ g++ goe.cpp
$ a.out
exception arised
```

- a) 5
- b) 10
- c) 15
- d) Positive Number Required

View Answer

Answer: a

Explanation: In this program, We are checking the age of a person, If it is zero means, We will arise a exception.

Output:


```
$ g++ goe1.cpp
$ a.out
5
```

- a) 50
- b) 0
- c) Division by zero condition!
- d) None of the mentioned

View Answer

Answer: c

Explanation: We are dividing the values and if one of the values is zero means, We are arising an exception.

Output:

```
$ g++ goe2.cpp
$ a.out
Division by zero condition!
```

- a) 15
- b) 5
- c) 2
- d) is not a valid operator

View Answer

Answer: d

Explanation: It will arise a exception because we missed a operator.

Output:

```
$ g++ goe3.cpp
$ a.out
is not a valid operator
```

- a) 5
- b) 2.236
- c) Error
- d) Cannot take sqrt of negative number

View Answer

Answer: b

Explanation: We are finding the square root of the number, if it is a positive number, it can manipulate, Otherwise it will arise a exception.

Output:

```
$ g++ goe4.cpp
$ a.out
2.236
```

9. How to handle the exception in constructor?

- a) We have to throw an exception
- b) We have to return the exception
- c) We have to throw an exception & return the exception
- d) none of the mentioned

View Answer

Answer: a

Explanation: As a constructor don't have a return type, We have to throw the exception.

10. What should present when throwing a object?

- a) constructor

b) copy-constructor
c) destructor
d) none of the mentioned
View Answer

Answer: b

Explanation: None.

ayzom.com

C++ Programming Questions and Answers – Resource Management

1. What can go wrong in resource management on c++?

- a) Leakage
- b) Exhaustion
- c) Dangling
- d) All of the mentioned

View Answer

Answer: d

Explanation: If there is any mishap in memory or resource management means, the problems that are mentioned above can happen.

2. When do we call that resource is leaked?

- a) Arise of compile time error
- b) It cannot be accessed by any standard mean
- c) Arise of runtime error
- d) None of the mentioned

View Answer

Answer: b

Explanation: Resource is said to be leaked when it cannot be accessed by any means of standard mean.

3. What kind of error can arise when there is a problem with memory?

- a) Segmentation fault
- b) Produce an error
- c) Both Segmentation fault & Produce an error
- d) none of the mentioned

View Answer

Answer: a

Explanation: segmentation fault error can arise when there is a problem with memory.

- a) 5
- b) 55
- c) 555
- d) Error: memory could not be allocated

View Answer

Answer: b

Explanation: As we had given i value as 2, It will print the 5 for two times.

Output:

```
$ g++ res.cpp
$ a.out
55
```

- a) Test
- b) TestTest
- c) Te
- d) None of the mentioned

View Answer

Answer: b

Explanation: We are copying the values from one variable to other, So it is printing is TestTest

Output:

```
$ g++ res1.cpp
$ a.out
TestTest
```

- a) 10
- b) 15
- c) error
- d) none of the mentioned

View Answer

Answer: a

Explanation: Even Though we passed the value, we didn't caught to manipulate it, So it is printing as 10.

Output:

```
$ g++ res2.cpp
$ a.out
10
```

- a) How r u
- b) segmentation fault
- c) error
- d) runtime error

View Answer

Answer: b

Explanation: As we are using a pointer value to copy a string, So it will be producing a runtime error.

Output:

```
$ g++ res3.cpp
$ a.out
segmentation fault
```

8. What is meant by garbage collection?

- a) The form of manual memory management
- b) The form of automatic memory management
- c) Used to replace the variables
- d) None of the mentioned

View Answer

Answer: b

Explanation: The garbage collection attempts to reclaim memory occupied by objects that are no longer in use by the program.

9. What are the operators available in C++ for dynamic allocation and de-allocation of memories?

- a) new
- b) delete
- c) compare
- d) both new & delete

View Answer

Answer: d

Explanation: new and delete operators are mainly used to allocate and deallocate during runtime.

10. Which is used to solve the memory management problem in c++?

- a) smart pointers
- b) arrays
- c) stack
- d) none of the mentioned

View Answer

Answer: a

Explanation: In C++, Smart pointers are used to manage memory issues like deallocate memory after use, checking bounds, etc.

ayzom.com

C++ Programming Questions and Answers – Exceptions That Are Not Errors

1. Which is used to handle the exceptions in c++?

- a) catch handler
- b) handler
- c) exception handler
- d) none of the mentioned

View Answer

Answer: c

Explanation: None.

2. Which type of program is recommended to include in try block?

- a) static memory allocation
- b) dynamic memory allocation
- c) const reference
- d) pointer

View Answer

Answer: b

Explanation: While during dynamic memory allocation, Your system may not have sufficient resources to handle it, So it is better to use it inside the try block.

3. Which statement is used to catch all types of exceptions?

- a) catch()
- b) catch(Test t)
- c) catch(...)
- d) none of the mentioned

View Answer

Answer: c

Explanation: This catch statement will catch all types of exceptions that arises in the program.

- a) -1
- b) 0
- c) Exception occurred: Thrown value is -1
- d) Error

View Answer

Answer: c

Explanation: As the given value is -1 and according to the condition, We are arising an exception.

Output:

```
$ g++ etae.cpp
$ a.out
Exception occurred: Thrown value is -1
```

- a) exception caught: std::bad_typeid
- b) exception caught: std::bad_alloc
- c) exception caught: std::bad_cast
- d) none of the mentioned

View Answer

Answer: a

Explanation: In this program, We used a bad type id for the polymorphic operator, So it is arising an bad_typeid exception.

Output:

```
$ g++ etae.cpp
$ a.out
exception caught: std::bad_typeid
```

- a) unexpected handler called
- b) caught bad_exception
- c) caught other exception
- d) both unexpected handler called & caught bad_exception

View Answer

Answer: d

Explanation: In this program, We are calling set_unexpected and myfunction, So it is printing the output as the given.

Output:

```
$ g++ etae.cpp
$ a.out
unexpected handler called
caught bad_exception
```

- a) -1
- b) ptr is NULL
- c) exception occurred: exiting
- d) none of the mentioned

View Answer

Answer: c

Explanation: catch(...) is used to catch all types of exceptions arising in the program.

Output:

```
$ g++ etea.cpp
$ a.out
Exception occurred: exiting
```

- a) caught other exception
- b) caught int
- c) unexpected called
- d) both caught int & unexpected called

View Answer

Answer: d

Explanation: As we are calling set_unexpected (myunexpected) function, this is printing as unexpected called and because of operator compliance it is arising an exception.

Output:

```
$ g++ etea.cpp
$ a.out
unexpected called
caught int
```

9. How to handle error in the destructor?

- a) throwing
- b) terminate
- c) both throwing & terminate
- d) none of the mentioned

View Answer

Answer: b

Explanation: It will not throw an exception from the destructor but it will the process by using terminate() function.

10. What kind of exceptions are available in c++?

- a) handled
- b) unhandled
- c) static
- d) dynamic

View Answer

Answer: b

Explanation: None.

ayzom.com

C++ Programming Questions and Answers – Exception Specifications

1. What is meant by exception specification?

- a) A function is limited to throwing only a specified list of exceptions
- b) A catch can catch all types of exceptions
- c) A function can throw any type of exceptions
- d) None of the mentioned

View Answer

Answer: a

Explanation: C++ provides a mechanism to ensure that a given function is limited to throwing only a specified list of exceptions. It is called as exception specification.

2. Identify the correct statement about throw(type).

- a) A function can throw any type of exceptions
- b) A function can throw an exception of certain type only
- c) A function can't throw any type of exception
- d) None of the mentioned

View Answer

Answer: b

Explanation: A function can throw an exception of certain type only.

3. What will happen when a programs throws any other type of exception other than specified?

- a) terminate
- b) arise an error
- c) run
- d) none of the mentioned

View Answer

Answer: b

Explanation: Because there is no way defined to catch that exception and as we know if an exception is not caught then error arises.

- a) In empty()
- b) Will throw an int
- c) Caught an int
- d) All of the mentioned

View Answer

Answer: d

Explanation: It will print all three because we are calling all functions in the main().

Output:

```
$ g++ exs.cpp
$ a.out
In empty()Will throw an intCaught an int
```

- a) NULL
- b) Exception:bad_alloc
- c) Exception:std::bad_typeid
- d) None of the mentioned

View Answer

Answer: c

Explanation: As we are using a bad type on pointers, So it is arising an error.

Output:

```
$ g++ exs1.cpp
$ a.out
Exception:std:bad_typeid
```

- a) out of range
- b) bad type_id
- c) bad allocation
- d) none of the mentioned

View Answer

Answer: a

Explanation: As we are using out of bound value on strings, So it arising an exception.

Output:

```
$ g++ exs2.cpp
$ a.out
Caught: basic_string::append
Type: Stl2out_of_range
#include <string>
```

- a) Can't do the dynamic_cast lor!!!
- b) Caught: bad_cast exception. Myshape is not mytriangle.
- c) Can't able to create the dynamic instance for the triangle, So it is arising an exception
- d) None of the mentioned

View Answer

Answer: c

Explanation: As we can't able to create the dynamic instance for the triangle, So it is arising an exception.

Output:

ayzom.com

advertisement

```
$ g++ exs3.cpp
$ a.out
Can't do the dynamic_cast lor!!!
Caught: bad_cast exception. Myshape is not mytriangle.
```

- a) 4
- b) 2
- c) bad_alloc
- d) depends on compiler

View Answer

Answer: d

Explanation: The size of unsigned long int always depends on compiler.

Output:

```
$ g++ exs4.cpp
$ a.out
4
```

9. What do you mean by "No exception specification"?

- a) It throws nothing
- b) It can throw anything
- c) It can catch anything
- d) None of the mentioned

View Answer

Answer: b

Explanation: None.

10. Which operations don't throw anything?

- a) Operations which are reversible
- b) Operations which are irreversible
- c) Operations which are static
- d) Operations which are dynamic

View Answer

Answer: b

Explanation: None.

ayzom.com

C++ Programming Questions and Answers – Uncaught Exceptions

1. What happens if try catch block is not used?

- a) arise an error
- b) program will run
- c) execute continuously
- d) none of the mentioned

View Answer

Answer: a

Explanation: If try catch block is not used the exception thrown by the program will be uncaught hence will result into error(s).

2. Which handler is used to handle all types of exception?

- a) catch handler
- b) catch-all handler
- c) catch-none handler
- d) none of the mentioned

View Answer

Answer: b

Explanation: To catch all types of exceptions, we use the catch-all handler.

3. Which operator is used as catch-all handler?

- a) ellipses operator
- b) ternary operator
- c) string operator
- d) unary operator

View Answer

Answer: a

Explanation: The ellipses operator can be represented as (...).

- a) Illegal initialization
- b) Terminate called after throwing an instance of 'int'
- c) Illegal initialization & terminate called after throwing an instance
- d) None of the mentioned

View Answer

Answer: b

Explanation: As we are throwing a negative number and we are using the only integer, So it is arising an error.

Output:

```
$ g++ uce.cpp
$ a.out
terminate called after throwing an instance of 'int'
```

- a) one
- b) inside catch
- c) one
terminate
- d) one
terminate
Aborted

View Answer

Answer: d

Explanation: This program uses set_terminate as it is having an uncaught exception.

Output:

```
$ g++ uce1.cpp
$ a.out
one
terminate
Aborted
```

- a) terminate handler called
- b) aborted
- c) both terminate handler & Aborted
- d) none of the mentioned

View Answer

Answer: c

Explanation: In this program, We are using set_terminate to abort the program.

Output:

```
$ g++ uce2.cpp
$ a.out
terminate handler called
Aborted
```

- a) Caught an exception
- b) NULL
- c) Both Caught an exception & NULL
- d) None of the mentioned

View Answer

Answer: a

Explanation: In this program, We are arising with the exception by using the method in the class.

Output:

advertisement

```
$ g++ uce3.cpp
$ a.out
Caught a exception
```

- a) Integer exception raised
- b) Funct() was called by terminate()
- c) Integer exception not raised
- d) none of the mentioned

View Answer

Answer: b

Explanation: As there is no integer in this program, We are printing Funct() was called by terminate().

Output:

```
$ g++ uce4.cpp
$ a.out
Funct() was called by terminate().
```

9. What function will be called when we have an uncaught exception?

- a) catch
- b) throw
- c) terminate
- d) none of the mentioned

View Answer

Answer: c

Explanation: If we have an uncaught exception means, the compiler will throw the control of the program to terminate function.

10. What will not be called when the terminate() is raised in the constructor?

- a) main()
- b) class
- c) destructor
- d) none of the mentioned

View Answer

Answer: c

Explanation: to free the memory occupied by that object during initializing and destroy that object.

ayzom.com

C++ Programming Questions and Answers – Exceptions and Efficiency

1. What will happen when we move to try block far away from catch block?

- a) Reduces the amount of code in cache
- b) Increases the amount of code in cache
- c) Don't alter anything
- d) None of the mentioned

View Answer

Answer: a

Explanation: compilers may try to move the catch-code far away from the try-code, which reduces the amount of code to keep in cache normally, thus enhancing performance.

2. What will happen if an exception that is thrown may cause a whole load of objects to go out of scope?

- a) Terminate the program
- b) Produce a runtime error
- c) It will be added to the overhead
- d) None of the mentioned

View Answer

Answer: c

Explanation: None.

3. What operation can be performed by destructor?

- a) Abort the program
- b) Resource cleanup
- c) Exit from the current block
- d) None of the mentioned

View Answer

Answer: b

Explanation: It will be used to free all the resources that are used by the block of code during execution.

- a) Memory allocated
- b) Exception arised
- c) Depends on the computer memory
- d) None of the mentioned

View Answer

Answer: c

Explanation: The value will be allocated, if there is enough memory in the system.

Output:

```
$ g++ expef.cpp
$ a.out
Memory allocated
```

- a) Catch a integer and that integer is:10
- b) Error
- c) Runtime error
- d) None of the mentioned

View Answer

Answer: c

Explanation: As the catch is created with a wrong type, So it will arise a runtime error.

Output:

```
$ g++ expref.cpp
$ a.out
Testing multiple catches
terminate called after throwing an instance of 'int'
:Aborted
```

- a) Invalid arguments
- b) Executed
- c) Error
- d) Runtime error

View Answer

Answer: b

Explanation: As we are throwing the function and catching it with a correct data type, So this program will execute.

Output:

```
$ g++ expref.cpp
$ a.out
Executed
```

- a) Exception is 5
- b) Exception is 1.1f
- c) Wrong number used
- d) None of the mentioned

View Answer

Answer: c

Explanation: As we are giving 3 to num, It is arising an exception named "wrong number used".

Output:

```
$ g++ expref.cpp
$ a.out
wrong number used
```

- a) 50
- b) 0
- c) Division by zero condition
- d) Error

View Answer

Answer: d

Explanation: As we missed the data type in the catch block, It will arise an error.

9. What is the main purpose of the constructor?

- a) Begin the execution of class
- b) Include the macros for the program
- c) Establish the class invariant
- d) None of the mentioned

View Answer

Answer: c

Explanation: The purpose of a constructor is to establish the class invariant. To do that, it often needs to acquire system resources or in general perform an operation that may fail.

10. Why is it expensive to use objects for the exception?

- a) Exception object is created only if an error actually happens
- b) Because of execution time
- c) Memory space involved in creating an exception object
- d) None of the mentioned

View Answer

Answer: a

Explanation: If an error occurs in the program, then only exception object is created otherwise, It will not be created. since throwing an exception triggers a bunch of actions during the stack unrolling, like invoking the the destructor of all the objects that has been created up to the point in which we are able to catch the exception, and invoking the destructor methods can imply flushing streams and freeing memory which can be expensive as well. Therefore, it's expensive to use in the program.

ayzom.com

C++ Programming Questions and Answers – Exception Handling – 1

1. What is an exception in C++ program?

- a) A problem that arises during the execution of a program
- b) A problem that arises during compilation
- c) Also known as the syntax error
- d) Also known as semantic error

View Answer

Answer: a

Explanation: An exception is defined as the problem in C++ program that arises during the execution of the program for example divide by zero error.

2. By default, what a program does when it detects an exception?

- a) Continue running
- b) Results in the termination of the program
- c) Calls other functions of the program
- d) Removes the exception and tells the programmer about an exception

View Answer

Answer: b

Explanation: By default, whenever a program detects an exception the program crashes as it does not know how to handle it hence results in the termination of the program.

3. Why do we need to handle exceptions?

- a) To avoid unexpected behaviour of a program during run-time
- b) To let compiler remove all exceptions by itself
- c) To successfully compile the program
- d) All of the mentioned

View Answer

Answer: a

Explanation: We need to handle exceptions in a program to avoid any unexpected behaviour during run-time because that behaviour may affect other parts of the program. Also, an exception is detected during run-time, therefore, a program may compile successfully even with some exceptions cases in your program.

4. How Exception handling is implemented in the C++ program?

- a) Using Exception keyword
- b) Using try-catch block
- c) Using Exception block
- d) Using Error handling schedules

View Answer

Answer: b

Explanation: C++ provides a try-catch block to handle exceptions in your program.

View Answer

Answer: a

Explanation: Try-catch block has the following syntax:

```
try{
    // codes that needs to check for exceptions
}
catch(Exception E1){
    // codes for handling exception....
    // Exception E denotes the type of exception this block is handling.
}
```

```
catch(Exception E2){  
    // other exception that needs to be handled...  
}
```

You can have any number of catch blocks catching different exceptions.....

6. Which part of the try-catch block is always fully executed?

- a) try part
- b) catch part
- c) finally part
- d) all of the mentioned

View Answer

Answer: c

Explanation: finally part of the try-catch block is always executed whether exceptions are caught or not.

7. Which of the following is an exception in C++?

- a) Divide by zero
- b) Semicolon not written
- c) Variable not declared
- d) An expression is wrongly written

View Answer

Answer: a

Explanation: Exceptions are those which are encountered during run-time of the program. semicolon, variable not declared and the wrong expression are compile-time errors, therefore, they are not exceptions. Divide by zero is the problem that is encountered during run-time, therefore, it is an exception.

8. What is an error in C++?

- a) Violation of syntactic and semantic rules of a languages
- b) Missing of Semicolon
- c) Missing of double quotes
- d) Violation of program interface

View Answer

Answer: a

Explanation: An error occurs when rules and laws of a language is violated while writing programs in that language.

9. What is the difference between error and exception?

- a) Both are the same
- b) Errors can be handled at the run-time but the exceptions cannot
- c) Exceptions can be handled at the run-time but the errors cannot
- d) Both can be handled during run-time

View Answer

Answer: c

Explanation: Exceptions can be handled during run-time whereas errors cannot be because exceptions occur due to some unexpected conditions during run-time whereas about errors compiler is sure and tells about them during compile-time.

10. What are the different types of exceptions?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: b

Explanation: There are two types of exceptions: Synchronous and asynchronous exceptions. Synchronous exceptions that are caused by the event which can be controlled by the program whereas Asynchronous exceptions are those which are beyond the control of the program.

11. Which keyword is used to throw an exception?

- a) try
 - b) throw
 - c) throws
 - d) except
- View Answer

Answer: b

Explanation: 'throw' keyword is used to throw exceptions if something bad happens.

- a) 0
 - b) 5
 - c) This value of b will make the product zero. So please provide positive values.
 - d) Product of 5 and 0 is: 0
- View Answer

Answer: c

Explanation: As the value of $b = 0$ is provided to the `func()` and the function is throwing an exception whenever the value of $b = 0$. Therefore the function throws the exception which will be printed on the screen.

Output:

```
$ ./a.out
This value of b will make the product zero. So please provide positive values.
```

- a) 0
 - b) Aborted (core dumped)
 - c) This value of b will make the product zero. So please provide positive values.
 - d) Product of 5 and 0 is: 0
- View Answer

Answer: b

Explanation: As the `func()` is throwing a `const char*` string but we the catch block is not catching any `const char*` exception i.e. exception thrown is not handled therefore the program results into Aborted(core dumped).

Output:

```
$ ./a.out
terminate called after throwing an instance of 'char const*'
Aborted (core dumped)
```

14. What is Re-throwing an exception means in C++?

- a) An exception that is thrown again as it is not handled by that catching block
 - b) An exception that is caught twice
 - c) An exception that is not handled in one caught hence thrown again
 - d) All of the mentioned
- View Answer

Answer: d

Explanation: Exception that is caught by a catch block but not handled by that catch block can be re-thrown by that catch block to further try-catch block.

- a) value of b is zero
- b) value of b is less than zero
- c) Product of 5 and -1 is: -5
- d) Aborted(core dumped)

View Answer

Answer: b

Explanation: Here the func() throws the value of b which is caught by the inner try-catch block, which again throws the message in order to handle different cases of b which is caught by the outer try-catch block. Now as the value of b is negative the program outputs the message as shown.

Output:

```
$ ./a.out  
value of b is less than zero
```

ayzom.com

C++ Programming Questions and Answers – Exception Handling – 2

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
#include <cstdlib>
using namespace std;
class A
{
    int a;
public:
    A() {}
};

class B: public A
{
    int b;
public:
    B() {}
};

void func()
{
    B b;
    throw b;
}

int main()
{
    try{
        func();
    }
    catch(A a){
        cout<<"Caught A Class\n";
    }
    catch(B b){
        cout<<"Caught B Class\n";
    }
}
```

1. Where should we place catch block of the derived class in a try-catch block?

- a) Before the catch block of Base class
- b) After the catch block of Base class
- c) Anywhere in the sequence of catch blocks
- d) After all the catch blocks

View Answer

Answer: a

Explanation: C++ asks the programmer to place the catch block of derived class before a catch block of the base class, otherwise derived catch block will never be executed.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
#include <cstdlib>
using namespace std;
class A
{
    int a;
public:
    A() {}
};

class B: public A
```

```

{
    int b;
public:
    B(){}
};

void func()
{
    B b;
    throw b;
}

int main()
{
    try{
        func();
    }
    catch(A a){
        cout<<"Caught A Class\n";
    }
    catch(B b){
        cout<<"Caught B Class\n";
    }
}

```

- a) The program compiles successfully without any errors or warnings
- b) Compile-time error occurs
- c) The program compiles successfully with warnings
- d) The program gives both errors and warnings

View Answer

Answer: c

Explanation: Catch block of derived should always be placed before the catch block base class, hence the program gives warnings stating that exceptions of the derived class will be caught by the base class.

Output:

```

$ g++ check.cpp
check.cpp: In function 'int main()':
check.cpp:33:2: warning: exception of type 'B' will be caught
    catch(B b){
    ^~~~~
check.cpp:30:2: warning:      by earlier handler for 'A'
    catch(A a){
    ^~~~~

```

3. What is the output of the following C++ code?

```

#include <iostream>
#include <string>
#include <cstdlib>
using namespace std;
class A
{
    int a;
public:
    A(){}
};

class B: public A
{
    int b;
public:
    B(){}
};

void func()
{
    B b;

```

```

        throw b;
    }

int main()
{
    try{
        func();
    }
    catch (A a) {
        cout<<"Caught A Class\n";
    }
    catch (B b) {
        cout<<"Caught B Class\n";
    }
}

```

- a) Caught B Class
 - b) Caught A Class
 - c) Compile-time error
 - d) Run-time error
- View Answer

Answer: b

Explanation: As the catch block of the derived class is after the catch block of base class, therefore, all the exceptions of the derived class will be caught by the base class, Hence the output of catch block of class A is printed.

Output:

```

$ ./a.out
Caught A Class

```

3. What is the output of the following C++ code?

```

#include <iostream>
#include <string>
#include <cstdlib>
using namespace std;
class A
{
    int a;
public:
    A() {}
};

class B: public A
{
    int b;
public:
    B() {}
};

void func()
{
    B b;
    throw b;
}

int main()
{
    try{
        func();
    }
    catch (A a) {
        cout<<"Caught A Class\n";
    }
    catch (B b) {
        cout<<"Caught B Class\n";
    }
}

```


- a) Caught B Class
- b) Caught A Class
- c) Compile-time error
- d) Run-time error

View Answer

Answer: a

Explanation: In this as the catch block of the derived class is before the catch block of the base class so when func() throws the object of class B it is caught by the catch block of class B, Hence the output is printed as shown.

Output:

```
$ ./a.out
Caught B Class
```

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
#include <cstdlib>
using namespace std;
class A
{
    int a;
public:
    A() {}
};

class B: public A
{
    int b;
public:
    B() {}
};

void func()
{
    B b;
    throw b;
}

int main()
{
    try{
        func();
    }
    catch(A a) {
        cout<<"Caught A Class\n";
    }
    catch(B b) {
        cout<<"Caught B Class\n";
    }
}
```

- a) Caught B Class
- b) Caught A Class
- c) Compile-time error
- d) Run-time error

View Answer

Answer: b

Explanation: The func() throws the object of class B but as catch block is defined to catch the exception of class B, Therefore the exception is caught by the base class A. The programmer has defined the catch block for B*, therefore, the object B is not caught by the pointer object B*.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
#include <cstdlib>
using namespace std;
class A
{
    int a;
public:
    A() {}
};

class B: public A
{
    int b;
public:
    B() {}
};

void func()
{
    B b;
    throw b;
}

int main()
{
    try{
        func();
    }
    catch(A a){
        cout<<"Caught A Class\n";
    }
    catch(B b){
        cout<<"Caught B Class\n";
    }
}
```

6. What id the syntax for catching any type of exceptions?

- a) catch(Exception e)
- b) catch(...)
- c) catch(Exception ALL)
- d) catch(ALL)

View Answer

Answer: b

Explanation: catch(...) is used in C++ to catch all types of exceptions in a single catch block.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
#include <cstdlib>
using namespace std;
class A
{
    int a;
public:
    A() {}
};

class B: public A
{
    int b;
public:
    B() {}
};

void func()
{
```

```

        B b;
        throw b;
    }

int main()
{
    try{
        func();
    }
    catch(A a){
        cout<<"Caught A Class\n";
    }
    catch(B b){
        cout<<"Caught B Class\n";
    }
}

```

View Answer

Answer: b

Explanation: Two try-catch blocks are declared each catching the respective exceptions from class A and B. But as we have defined catch all exceptions in the first case, therefore, the exception for class B is caught when thrown by the func1(), but in the second case, the try-catch block is catching only the exception for class B so when func2() throws class A exception and no catch block to catch that exception therefore program results into abort(core dumped).

3. What is the output of the following C++ code?

```

#include <iostream>
#include <string>
#include <cstdlib>
using namespace std;
class A
{
    int a;
public:
    A(){}
};

class B: public A
{
    int b;
public:
    B(){}
};

void func()
{
    B b;
    throw b;
}

int main()
{
    try{
        func();
    }
    catch(A a){
        cout<<"Caught A Class\n";
    }
    catch(B b){
        cout<<"Caught B Class\n";
    }
}

```

8. Uncaught exception leads to _____

- a) termination of program
- b) successful execution of programs
- c) no effect on the program
- d) execution of other functions of the program starts

View Answer

Answer: a

Explanation: Uncaught exceptions in a program leads to the termination of a program.

3. What is the output of the following C++ code?

```

#include <iostream>
#include <string>
#include <cstdlib>
using namespace std;
class A
{
    int a;
public:
    A() {}
};

class B: public A
{
    int b;
public:
    B() {}
};

void func()
{
    B b;
    throw b;
}

int main()
{
    try{
        func();
    }
    catch(A a) {
        cout<<"Caught A Class\n";
    }
    catch(B b) {
        cout<<"Caught B Class\n";
    }
}

```

9. An uncaught handler returns to _____

- a) main function
- b) its caller
- c) its callee
- d) waits there for some time

View Answer

Answer: c

Explanation: Uncaught handler returns to its callee(i.e. the function it is called by).

3. What is the output of the following C++ code?

```

#include <iostream>
#include <string>
#include <cstdlib>
using namespace std;
class A
{
    int a;
public:
    A() {}
};

```

```
class B: public A
{
    int b;
public:
    B() {}
};

void func()
{
    B b;
    throw b;
}

int main()
{
    try{
        func();
    }
    catch (A a) {
        cout<<"Caught A Class\n";
    }
    catch (B b) {
        cout<<"Caught B Class\n";
    }
}
```

10. Header file used for exception handling in C++?

- a) <cstdlib>
- b) <string>
- c) <handler>
- d) <exception>

View Answer

Answer: d

Explanation: <exception> header file is used to use exception handler in C++.

C++ Programming Questions and Answers – Exception Handling – 3

1. The C++ code which causes abnormal termination/behaviour of a program should be written under _____ block.

- a) try
- b) catch
- c) finally
- d) throw

View Answer

Answer: a

Explanation: Code that leads to the abnormal termination of the program should be written under the try block.

2. Exception handlers are declared with _____ keyword.

- a) try
- b) catch
- c) throw
- d) finally

View Answer

Answer: b

Explanation: C++ uses catch block to handle any exceptions that occur during run-time of the program.

3. Which of the following statements are correct about Catch handler?

- i. It must be placed immediately after the try block
- ii. It can have more than one parameters
- iii. There must be one and only one catch handler for every try block
- iv. There can be multiple catch handler for a try block
- v. General catch handler can be kept anywhere after try block.

- a) i, iv, v
- b) i, ii, iii
- c) i, iv
- d) i, ii

View Answer

Answer: c

Explanation: A catch block should always be placed after the try block and there can be multiple catch block following a try block.

4. In nested try-catch block, if the inner catch block gets executed, then _____

- a) Program stops immediately
- b) Outer catch block also executes
- c) Compiler jumps to the outer catch block and executes remaining statements of the main() function
- d) Compiler executes remaining statements of outer try-catch block and then the main() function

View Answer

Answer: d

Explanation: The inner catch block will be executed then remaining part of the outer try block will be executed and then the main block will be executed.

5. If inner catch block is unable to handle the exception thrown then _____

- a) The compiler looks for the outer try-catch block
- b) Program stops abnormally
- c) The compiler will check for appropriate catch handler of the outer try block
- d) All of the mentioned

View Answer

Answer: c

Explanation: In such cases, the compiler will try to find an appropriate outer catch block to handle the exception otherwise if nothing is there then occurs the abnormal behaviour of the program.

6. In nested try catch blocks, if both inner and outer catch blocks are unable to handle the exception thrown, then _____

- a) Compiler executes only main()
- b) Compiler throws compile time errors about it
- c) Program will run without any interrupt
- d) Program will be terminated abnormally

View Answer

Answer: d

Explanation: If no inner/outer catch handler is available to handle the exception then as usual the program will show abnormal behaviour.

7. Which function is invoked when an unhandled exception is thrown?

- a) stop()
- b) aborted()
- c) terminate()
- d) abandon()

View Answer

Answer: c

Explanation: terminate() function is called/invoked in case any exception is not handled properly.

8. How one can restrict a function to throw particular exceptions only?

- a) By defining multiple try-catch blocks inside a function
- b) By defining a generic function within a try-catch block
- c) By defining a function with throw clauses
- d) Not allowed in C++

View Answer

Answer: c

Explanation: We can use throw clause to mention the exceptions that a function can throw. Hence restricting the function to throw some particular exceptions only.

9. Which function is invoked when we try to throw an exception that is not supported by a function?

- a) indeterminate()
- b) unutilized()
- c) unexpected()
- d) unpredicted()

View Answer

Answer: c

Explanation: As the exception is not supported by the function so it does not know what to do about the exception in that case it calls the unexpected() function of the STL library.

10. Return type of uncaught_exception() is _____

- a) int
- b) bool
- c) char *
- d) double

View Answer

Answer: b

Explanation: Return type of uncaught exceptions are bool.

View Answer

Answer: a

Explanation: "Inside try" will always be printed as we just entering try block then. Now as $var < 0$ therefore the try block will throw int var as exception hence "After throw" will not be printed) Now this exception will be caught by the catch handler printing "Exception caught" and at last after terminating the program "After catch" will be printed.

c) Error

d) Run-time error

View Answer

Answer: d

Explanation: As no catch handler is defined to catch an integer hence when var variable, which is int, is thrown then nothing is there to catch the int hence the program terminates abnormally.

d) Error

View Answer

Answer: a

Explanation: As exception thrown by the inner try block is caught by the inner catch block hence the exception is handled at the inner level and program continues to run outer try block, statement afterwards.

d) Error

View Answer

Answer: b

Explanation: As there is no inner catch handler to handle the int exception thrown by the try block therefore outer catch block handler catches the exception thrown by the inner try catch therefore the output prints "Outer Catch" instead of "Inner Catch". After that program continues execution.

d) Error

View Answer

Answer: c

Explanation: The exception thrown by the inner try catch block is caught by the inner block hence "Inner Catch" is printed but as inner catch block again throws an exception further therefore the exception is thrown further which is caught by the outer catch block hence "Outer Catch" is also printed.

15. Which of the following is true about exception handling in C++?

i) There is a standard exception class in C++ similar to Exception class in Java.

ii) All exceptions are unchecked in C++, i.e., the compiler does not checks if the exceptions are caught or not.

iii) In C++, a function can specify the list of exceptions that it can throw using comma separated list like following.

void fun(int a, char b) throw (Exception1, Exception2, ..)

a) i, iii

b) i, ii, iii

c) i, ii

d) ii, iii

View Answer

Answer: b

Explanation: In C++ also we have an exception class similar to java. All exceptions are unchecked in C++. We can specify the list of exception that a function throws using the above format.

C++ Programming Questions and Answers – Error Handling Alternatives

1. Which alternative can replace the throw statement?

- a) for
- b) break
- c) return
- d) exit

View Answer

Answer: c

Explanation: throw and return does the same job as return a value. So it can be replaced.

2. What are the disadvantages if use return keyword to return error codes?

- a) You have to handle all exceptional cases explicitly
- b) Your code size increases dramatically
- c) The code becomes more difficult to read
- d) All of the mentioned

View Answer

Answer: d

Explanation: As we are using return for each and every exception, It will definitely increase the code size.

3. What is most suitable for returning the logical errors in the program?

- a) Use constructor and destructor
- b) Set a global error indicator
- c) Use break keyword
- d) None of the mentioned

View Answer

Answer: b

Explanation: None.

a) c

1A

b) x

c) Both c & x

d) None of the mentioned

View Answer

Answer: a

Explanation: We are checking the type id of char and float as they are not equal, We are printing c.

Output:

```
$ g++ eal.cpp
$ a.out
c
1A
```

a) 0

b) Bad operator

c) 10

d) None of the mentioned

View Answer

Answer: a

Explanation: We are dividing 0 and 10 in this program and we are using the throw statement in the function block.

Output:

```
$ g++ eal.cpp
$ a.out
0
```

- a) 256
- b) Invalid argument
- c) Error
- d) None of the mentioned

View Answer

Answer: c

Explanation: We can't return a statement by using the return keyword, So it is arising an error.

7. What is the use of RAII in c++ programming?

- a) Improve the exception safety
- b) Terminate the program
- c) Exit from the block
- d) None of the mentioned

View Answer

Answer: a

Explanation: None.

8. How many levels are there in exception safety?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: c

Explanation: The three levels of exception safety are basic, strong and no throw.

9. Pick out the correct statement for error handling alternatives.

- a) Terminate the program
- b) Use the stack
- c) Exit from the block
- d) None of the mentioned

View Answer

Answer: b

Explanation: When an error is raised means, it will be pushed into stack and it can be corrected later by the programmer.

10. What will happen when an exception is not processed?

- a) It will eat up a lot of memory and program size
- b) Terminate the program
- c) Crash the compiler
- d) None of the mentioned

View Answer

Answer: a

Explanation: As in the case of not using an exception, it will remain useless in the program and increase the code complexity.

C++ Programming Questions and Answers – Standard Exceptions

View Answer

Answer: a

Explanation: None.

2. Where are standard exception classes grouped?

- a) namespace std
- b) error
- c) catch
- d) none of the mentioned

View Answer

Answer: a

Explanation: As these are standard exceptions, they need to be defined in the standard block, So it is defined under namespace std.

3. How many types of standard exception are there in c++?

- a) 9
- b) 5
- c) 6
- d) 7

View Answer

Answer: a

Explanation: There are nine standard exceptions in c++. They are `bad_alloc`, `bad_cast`, `bad_exception`, `bad_function_call`, `bad_typeid`, `bad_weak_ptr`, `ios_base::failure`, `logic_error` and `runtime_error`.

- a) My
- b) My exception
- c) No exception
- d) None of the mentioned

View Answer

Answer: b

Explanation: This is a type of exception arising in the class. We can call this also as a standard exception.

Output:

```
$ g++ std.cpp
$ a.out
My exception
```

- a) Allocated
- b) Standard exception:
- c) `bad_alloc`
- d) Depends on memory

View Answer

Answer: d

Explanation: Variable will be allocated depends on the available space in the memory, If there is no space means, It will throw an exception.

Output:

```
$ g++ std1.cpp
$ a.out
Allocated
```

- a) 0
- b) 2
- c) bad_alloc
- d) depends on compiler

View Answer

Answer: a

Explanation: As we are dividing the zero by three, it is returning 0.

Output:

```
$ g++ std2.cpp
$ a.out
0
```

-
- a) Caught standard exception
 - b) No exception arises
 - c) Caught: bad_cast exception
 - d) None of the mentioned

View Answer

Answer: c

Explanation: As we are not able to allocate the values by using dynamic cast, So it is arising an exception.

Output:

```
$ g++ std3.cpp
$ a.out
Caught: bad_cast exception
```

-
- a) No exception arises
 - b) The object is null
 - c) Error
 - d) None of the mentioned

View Answer

Answer: b

Explanation: As there is no object in the class, It is arising an exception in the program.

Output:

```
$ g++ std4.cpp
$ a.out
The object is null
```

9. Which of the following is best to include under try block?

- a) static values
- b) const values
- c) dynamic allocations
- d) none of the mentioned

View Answer

Answer: c

Explanation: Because the dynamic allocations can change at any time, So it is best to include in try block.

10. What are the predefined exceptions in c++?

- a) Memory allocation errors
- b) I/O errors
- c) Both Memory allocation errors & I/O errors
- d) None of the mentioned

View Answer

Answer: c

Explanation: None.

ayzom.com

C++ Programming Questions and Answers – Class Hierarchies Introduction

1. What will happen when introduce the interface of classes in a run-time polymorphic hierarchy?

- a) Separation of interface from implementation
- b) Merging of interface from implementation
- c) Separation of interface from debugging
- d) None of the mentioned

View Answer

Answer: a

Explanation: None.

2. Which classes are called as mixin?

- a) Represent a secondary design
- b) Classes express functionality which represent responsibilities
- c) Standard logging stream
- d) None of the mentioned

View Answer

Answer: b

Explanation: A class that expresses functionality rather than its primary design role is called a mixin.

3. What is the use of clog?

- a) Standard logging stream
- b) Error stream
- c) Input stream
- d) None of the mentioned

View Answer

Answer: a

Explanation: clog is an object of class ostream that represents the standard logging stream. It is associated with the cstdio stream stderr, like cerr.

- a) 5593.54
- b) Error
- c) Runtime error
- d) None of the mentioned

View Answer

Answer: a

Explanation: In this program, We have used the string hierarchy to compute the square of the number.

Output:

```
$ g++ class.cpp
$ a.out
The double value is : 5593.54 .
```

- a) The program executed
- b) Error
- c) Runtime error
- d) None of the mentioned

View Answer

Answer: b

Explanation: We are allowed to overload constructor but in this case as both the constructor have no parameters which implies that both the constructor have same signature which is not allowed i.e. constructors can be overloaded but two overloaded constructors can not have same

function signature.

- a) Caught Base Class Exception
- b) Caught Derived Class Exception
- c) Caught Base & Derived Class Exception
- d) None of the mentioned

View Answer

Answer: b

Explanation: As we are throwing the value from the derived class, it is arising an exception in derived class

Output:

advertisement

```
$ g++ class1.cpp
$ a.out
Caught Derived Class Exception
```

- a) a long* string
- b) a long st*ring
- c) Depends on compiler
- d) None of the mentioned

View Answer

Answer: c

Explanation: In this program, We are placing the string based on the size of the string and it is a string hierarchy.

Output:

```
$ g++ class2.cpp
$ a.out
a long* string
```

8. How many types of guarantees are there in exception class can have?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: c

Explanation: There are three types of guarantees in c++. They are weak, strong and no-throw.

9. Which operator is used to create the user-defined streams in c++?

- a) >>
- b) <<
- c) &
- d) Both >> & <<

View Answer

Answer: d

Explanation: We can make user-defined types with streams by overloading the insertion operator (<<) to put objects into streams and the extraction operator (>>) to read objects from streams.

10. What does the cerr represent?

- a) Standard error stream
- b) Standard logging stream
- c) Input stream

d) Output stream

View Answer

Answer: a

Explanation: cerr is an object of class ostream that represents the standard error stream. It is associated with the cstdio stream stderr.

ayzom.com

C++ Programming Questions and Answers – Multiple Inheritance

1. What is meant by multiple inheritance?

- a) Deriving a base class from derived class
- b) Deriving a derived class from base class
- c) Deriving a derived class from more than one base class
- d) None of the mentioned

View Answer

Answer: c

Explanation: Multiple inheritance enables a derived class to inherit members from more than one parent.

2. Which symbol is used to create multiple inheritance?

- a) Dot
- b) Comma
- c) Dollar
- d) None of the mentioned

View Answer

Answer: b

Explanation: For using multiple inheritance, simply specify each base class (just like in single inheritance), separated by a comma.

3. Which of the following advantages we lose by using multiple inheritance?

- a) Dynamic binding
- b) Polymorphism
- c) Both Dynamic binding & Polymorphism
- d) None of the mentioned

View Answer

Answer: c

Explanation: The benefit of dynamic binding and polymorphism is that they help making the code easier to extend but by multiple inheritance it makes harder to track.

- a) 20
- b) 10
- c) 20
10
- d) None of the mentioned

View Answer

Answer: c

Explanation: We are using the multiple inheritance to find the area of rectangle and triangle.

Output:

```
$ g++ mul.cpp
$ a.out
20
10
```

- a) DerivedOne
- b) DerivedTwo
- c) Error
- d) None of the mentioned

View Answer

Answer: c

Explanation: In this program, 'Base' is an ambiguous base of 'Multiple'. So it is producing an error. And this program is a virtual base class.

- a) 3100
- b) 3010
- c) 2010
- d) 1010

View Answer

Answer: b

Explanation: In this program, We are calculating the total and average marks of a student by using multiple inheritance.

Output:

advertisement

```
$ g++ mul1.cpp
$ a.out
3010
```

- a) Inherited
- b) Error
- c) Runtime error
- d) None of the mentioned

View Answer

Answer: a

Explanation: In this program, We apply the multiple inheritance to structure.

Output:

```
$ g++ mul2.cpp
$ a.out
Inherited
```

- a) 100
- b) 200
- c) Both 100 & 200
- d) None of the mentioned

View Answer

Answer: c

Explanation: In this program, We are passing the values by using multiple inheritance and printing the derived values.

Output:

```
$ g++ mul4.cpp
$ a.out
100
200
```

9. Which design patterns benefit from the multiple inheritances?

- a) Adapter and observer pattern
- b) Code pattern
- c) Glue pattern
- d) None of the mentioned

View Answer

Answer: a

Explanation: None.

10. What are the things are inherited from the base class?

- a) Constructor and its destructor
- b) Operator=() members
- c) Friends
- d) All of the mentioned

View Answer

Answer: d

Explanation: These things can provide necessary information for the base class to make a logical decision.

ayzom.com

C++ Programming Questions and Answers – Inheritance – 1

1. What is Inheritance in C++?

- a) Wrapping of data into a single class
- b) Deriving new classes from existing classes
- c) Overloading of classes
- d) Classes with same names

View Answer

Answer: b

Explanation: Inheritance is the concept of OOPs in which new classes are derived from existing classes in order to reuse the properties of classes defined earlier.

2. How many specifiers are used to derive a class?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: c

Explanation: There are 3 specifiers used to derive a class. They are private, protected and public.

3. Which specifier makes all the data members and functions of base class inaccessible by the derived class?

- a) private
- b) protected
- c) public
- d) both private and protected

View Answer

Answer: a

Explanation: Private access specifier is used to make all the data members and functions of the base class inaccessible.

4. If a class is derived privately from a base class then _____

- a) no members of the base class is inherited
- b) all members are accessible by the derived class
- c) all the members are inherited by the class but are hidden and cannot be accessible
- d) no derivation of the class gives an error

View Answer

Answer: c

Explanation: Whenever a class is derived, all the members of the base class is inherited by the derived class but are not accessible by the derived class.

- a) 8
- b) 12
- c) Error
- d) Segmentation fault

View Answer

Answer: b

Explanation: As class B is derived from class A and class A has three members with each of 4 bytes size hence size of B equal to $3 * 4 = 12$ bytes.

- a) 1010

b) 1510

c) 1515

d) 5110

View Answer

Answer: b

Explanation: When change() is called it sets parents class 'a' variable = 10. When print() is called then 'a' from class B is printed and when value_of_a() is called then 'a' from class A is printed.

View Answer

Answer: a

Explanation: When a derived class is declared it calls both its constructor and the base class constructor. It first calls the base class constructor and then its own constructor.

8. What is a virtual function in C++?

a) Any member function of a class

b) All functions that are derived from the base class

c) All the members that are accessing base class data members

d) All the functions which are declared in the base class and is re-defined/overridden by the derived class

View Answer

Answer: d

Explanation: Virtual function is a function that is declared inside the base class and is re-defined inside the derived class.

9. Which is the correct syntax of declaring a virtual function?

a) virtual int func();

b) virtual int func() {};

c) inline virtual func();

d) inline virtual func() {};

View Answer

Answer: a

Explanation: To make a function virtual function we just need to add virtual keyword at the starting of the function declaration.

a) Hello this is class B

b) Hello this is class A

c) Error

d) Segmentation fault

View Answer

Answer: a

Explanation: Normal execution of the program and object calls func() from class B.

a) Hello this is class A

b) Hello this is class B

c) Error

d) Segmentation Fault

View Answer

Answer: d

Explanation: As object 'a' is a pointer object and we know every pointer needs to be initialised memory before use. Hence segmentation fault. Use A *a = new A(); to initialise memory to the object.

a) Hello this is class A

b) Hello this is class B

c) Error

d) Segmentation Fault

View Answer

Answer: b

Explanation: As pointer object a is pointing to the object b hence the definition of virtual function defined inside the class B will be class. This is one of the use of virtual function.

13. Which statement is incorrect about virtual function.

- a) They are used to achieve runtime polymorphism
- b) They are used to hide objects
- c) Each virtual function declaration starts with the virtual keyword
- d) All of the mentioned

View Answer

Answer: b

Explanation: Virtual function are used to achieve runtime polymorphism by calling the right function during runtime. Their declaration starts with a virtual keyword.

14. The concept of deciding which function to invoke during runtime is called _____

- a) late binding
- b) dynamic linkage
- c) static binding
- d) both late binding and dynamic linkage

View Answer

Answer: d

Explanation: The concept of deciding which function to invoke during runtime is called late binding or dynamic linkage. Late binding because function binding to the object is done during runtime. Dynamic linkage because this binding is done during runtime.

15. What is a pure virtual function?

- a) A virtual function defined inside the base class
- b) A virtual function that has no definition relative to the base class
- c) A virtual function that is defined inside the derived class
- d) Any function that is made virtual

View Answer

Answer: b

Explanation: A virtual function that has no definition relative to the base class is called a pure virtual function.

C++ Programming Questions and Answers – Inheritance – 2

View Answer

Answer: a

Explanation: As the Male class is derived from Human class and Human class is derived from the Mammal class. Therefore when an object of Male is declared then three constructors will be called namely Mammal(), Human() and Male() in the given order.

2. What is the order of Constructors call when the object of derived class B is declared, provided class B is derived from class A?

- a) Constructor of A followed by B
- b) Constructor of B followed by A
- c) Constructor of A only
- d) Constructor of B only

View Answer

Answer: a

Explanation: Firstly the Constructor of class A is called then class B because the Constructor of the base class is called before derived class.

3. What is the order of Destructors call when the object of derived class B is declared, provided class B is derived from class A?

- a) Destructor of A followed by B
- b) Destructor of B followed by A
- c) Destructor of A only
- d) Destructor of B only

View Answer

Answer: b

Explanation: Order of Destructor call is just reverse of the order of Constructors call. First, the destructor of the derived class is called then Destructor of the base class is called.

- a) I'm a Male
- b) I'm a Mammal
- c) Error
- d) Segmentation Fault

View Answer

Answer: d

Explanation: As the Mammal pointer *M is not Initialized memory therefore program results into segmentation faults.

View Answer

Answer: c

Explanation: There is a difference between pointer and references. Pointer stores the address of a variable so we need dereferencing operator to access the pointed variable whereas references are another name for that variable so we don't need any dereferencing operator, they are dereference by compiler itself therefore when we are using pointer then Mammal class definition is called and when reference is used then Male class definition is used.

6. Virtual functions in C++ tells the compiler to perform _____ on such functions.

- a) static binding
- b) late binding
- c) compile time binding
- d) no binding

View Answer

Answer: b

Explanation: Virtual function in C++ adds the power of late binding by deciding the type of object during run-time.

- a) Error

- b) Segmentation fault
 - c) I'm a Human
 - d) Garbage Value
- View Answer

Answer: c

Explanation: Using base class pointer we can call private functions of derived by using virtual keyword because virtual function asks compiler performs late binding i.e. bind function at run-time and at run-time there is no checking of access specifiers. Hence it can access private members.

8. Which concept of OOPs is shown by Virtual Functions?

- a) Inheritance
 - b) Encapsulation
 - c) Polymorphism
 - d) Abstraction
- View Answer

Answer: c

Explanation: Virtual function allows us to give different definitions of the same function i.e. overloading of functions which is known as Polymorphism.

-
- a) A's Constructor
 - b) Present inside the class B
 - c) Error
 - d) Segmentation fault
- View Answer

Answer: c

Explanation: C++ does not allow programmers to make constructor a virtual function.

-
- a) class B
 - b) Error
 - c) Segmentation fault
 - d) class A
- View Answer

Answer: b

Explanation: A function cannot be made virtual and static at the same time.

-
- a) 1
 - b) 0
 - c) Segmentation fault
 - d) Error
- View Answer

Answer: d

Explanation: Non-static members of class cannot be used inside a static function of class.

-
- a) 1
 - b) 0
 - c) Error
 - d) Segmentation fault
- View Answer

Answer: a

Explanation: Though the constructor of class A is not called to initialize variable 'a' but as we know whenever we create an object of the derived class, constructors of both base and derived classes are called hence variable 'a' is initialized and program runs perfectly.

- a) Error
- b) Segmentation fault
- c) a: 1
- d) a: 0

View Answer

Answer: a

Explanation: As class B is derived privately from A hence all the members of class A cannot be accessible by the object of class B hence the program gives error.

- a) Error because of the conflicts between two show() function in class B
- b) Program will compile successfully
- c) Error due to self call in show() function
- d) Error because show() function from class A is derived privately

View Answer

Answer: b

Explanation: As the program is syntactically correct and as one show() function is in class A and other in class B therefore no conflicts in same name function. Therefore program compiles successfully.

15. Pick the correct statement.

- a) Virtual function can have different names in the base and derived class
- b) Virtual function cannot be applied in Multiple Inheritance classes
- c) Virtual function are different in definitions only
- d) Virtual function does early binding

View Answer

Answer: c

Explanation: Virtual functions differ in definitions only, prototype are similar. They does the late binding. They are applicable to all types of inheritance.

C++ Programming Questions and Answers – Access Control

1. Which access specifier is use where one wants data members to be accessed by other classes but not from outside objects?

- a) private
- b) protected
- c) public
- d) both protected and public

View Answer

Answer: b

Explanation: Protected and public members are accessible from derived classes but public members can be accessed by objects of the class so protected specifier is the answer.

2. Which of the following describes the protected access specifier?

- a) The variable is visible only outside inside the block
- b) The variable is visible everywhere
- c) The variable is visible to its block and to it's derived class
- d) None of the mentioned

View Answer

Answer: c

Explanation: Protected members are visible to its block and to the derived classes and not visible to outside objects or variables.

3. To which of the following access specifiers are applicable?

- a) Member data
- b) Functions
- c) Both Member data & Functions
- d) None of the mentioned

View Answer

Answer: c

Explanation: The access specifiers can be applicable to the member data and functions because they need to be accessed outside the block.

- a) 3010
- b) 1010
- c) 2100
- d) Error

View Answer

Answer: a

Explanation: In this program we setting values of m1 and m2 using obj.setObject() function derived from student class. setting value of sm using getsm() derived from sports function and then displaying the outputs using display() function in statement class.

- a) Allocated
- b) Error
- c) 3.14159
- d) None of the mentioned

View Answer

Answer: a

Explanation: In this program, We used access specifiers for structures, As we declared all methods as public, The values can be allocated.

Output:

```
$ g++ acc.cpp
$ a.out
Allocated
```

- a) 50
 - b) Identical results would be produced
 - c) Error
 - d) Runtime error
- View Answer

Answer: b

Explanation: In this program, We apply the access specifiers to both the class and the structure.

Output:

```
$ g++ acc1.cpp
$ a.out
```

Identical results would be produced

- a) Gates is
 - b) Gates is 56 years old
 - c) Error
 - d) None of the mentioned
- View Answer

Answer: b

Explanation: In this program, We passed the value from main function to class and returning it to the main and then printing it.

Output:

```
$ g++ acc2.cpp
$ a.out
Gates is 56 years old
```

advertisement

- a) 99
 - b) 47
 - c) Data accessed
 - d) None of the mentioned
- View Answer

Answer: c

Explanation: In this program, We are using the access specifiers to friend function to manipulate the values.

Output:

```
$ g++ acc3.cpp
$ a.out
Data accessed
```

9. Members of which access specifiers are not inherited?

- a) Public
 - b) Protected
 - c) Private
 - d) None of the mentioned
- View Answer

Answer: d

Explanation: All the data members and member functions of a class are private by default.

10. What is the importance of mutable keyword?

- a) It allows the data member to change within a const member function

- b) It will not allow the data member to change within a const member function
- c) It will copy the values of the variable
- d) None of the mentioned

View Answer

Answer: a

Explanation: Mutable keyword allows assigning values to a data member belonging to a class defined as “Const” or constant.

ayzom.com

C++ Programming Questions and Answers – Run Time Type Information

1. What is the Run-Time Type Information?

- a) Information about an object's data type at runtime
- b) Information about the variables
- c) Information about the given block
- d) None of the mentioned

View Answer

Answer: a

Explanation: With the help of RTTI, We can get the information about the data type at the runtime.

2. Which operators are part of RTTI?

- a) `dynamic_cast()`
- b) `typeid`
- c) both `dynamic_cast()` & `typeid`
- d) none of the mentioned

View Answer

Answer: c

Explanation: The `dynamic_cast` operation and `typeid` operator in C++ are part of RTTI.

3. To which type of class, We can apply RTTI?

- a) Encapsulation
- b) Polymorphic
- c) Derived
- d) None of the mentioned

View Answer

Answer: b

Explanation: RTTI is available only for classes which are polymorphic, which means they have at least one virtual method.

- a) Null pointer on first type-cast
- b) Null pointer on second type-cast
- c) Exception
- d) None of the mentioned

View Answer

Answer: b

Explanation: In this program, We apply the dynamic cast to `pd`. Based on the value in the `pd`, it produces the output.

Output:

```
$ g++ rtti.cpp
$ a.out
Null pointer on second type-cast
```

- a) `Pi`
- b) `i`
- c) Both `pi` & `i`
- d) `f`

View Answer

Answer: c

Explanation: In this program, We are finding the `typeid` of the given variables.

Output:

```
$ g++ rtti1.cpp
$ a.out
Pii
```

- a) base*
- b) derived*
- c) 4base and 7derived
- d) none of the mentioned

View Answer

Answer: c

Explanation: In this program, We apply the typeid to the polymorphic class.

Output:

```
$ g++ rtti2.cpp
$ a.out
4base      7derived
```

- a) int
- b) float
- c) double
- d) object is NULL

View Answer

Answer: d

Explanation: In this program, We are using the bad typeid() for a. So it is arising an exception.

Output:

```
$ g++ rtti3.cpp
$ a.out
object is NULL
```

- a) Class C
- b) Class A
- c) Both Class C & A
- d) None of the mentioned

View Answer

Answer: c

Explanation: In this program, We applied the dynamic casting to structure and produced the output.

Output:

```
$ g++ rtti4.cpp
$ a.out
Class C
Class A
```

9. What is meant by type_info?

- a) Used to hold the type information returned by the typeid operator
- b) Used to hold the type information returned by the dynamic_cast
- c) Used to hold the type information returned by the static_cast
- d) None of the mentioned

View Answer

Answer: a

Explanation: None.

10. At which time does the static_cast can be applied?

- a) Compile-time construct
- b) Runtime construct
- c) Both Compile-time & Runtime construct
- d) None of the mentioned

View Answer

Answer: a

Explanation: Static_cast can be applied to only compile-time construct and not during run time construct.

ayzom.com

C++ Programming Questions and Answers – Pointers to Members

1. Which is referred by pointers to member?

- a) static members of class objects
- b) Non-static members of class objects
- c) Referring to whole class
- d) None of the mentioned

View Answer

Answer: b

Explanation: We cannot use a pointer to member to point to a static class member because the address of a static member is not associated with any particular object.

2. What should be used to point to a static class member?

- a) Smart pointer
- b) Dynamic pointer
- c) Normal pointer
- d) None of the mentioned

View Answer

Answer: c

Explanation: None.

3. Which operator is used in pointer to member function?

- a) *
- b) ->*
- c) Both * & ->*
- d) None of the mentioned

View Answer

Answer: c

Explanation: The pointer to member operators * and ->* are used to bind a pointer to a member of a specific class object.

a) 10

20

b) 20

10

c) 20

d) 10

View Answer

Answer: a

Explanation: In this program, We are assigning 10 and printing it in the main function and then for value 20, We are passing the value to class and printing it.

Output:

```
$ g++ ptm.cpp
$ a.out
10
20
```

a) func1

b) func1

func1

c) 1

2

d) func1

func1

1

2

View Answer

Answer: d

Explanation: In this program, As we are passing the value twice to the method in the class, It is printing the func1 twice and then it is printing the given value.

Output:

```
$ g++ ptm1.cpp
$ a.out
func1
func1
1
2
```

a) 1

b) 2

c) Both 1 & 2

d) None of the mentioned

View Answer

Answer: c

Explanation: In this program, We are printing the value by direct access and another one by using pointer to member.

Output:

```
$ g++ ptm2.cpp
$ a.out
1
2
```

a) I have 4 apples

I have 7 oranges

b) I have 3 apples

I have 5 oranges

c) I have 1 apples

I have 5 oranges

d) None of the mentioned

View Answer

Answer: a

Explanation: In this program, We are passing the value to the class and adding the values and printing it in the main.

Output:

advertisement

```
$ g++ ptm3.cpp
$ a.out
I have 4 apples
I have 7 oranges
```

a) Executed

b) Error

c) Runtime error

d) None of the mentioned

View Answer

Answer: a

Explanation: In this program, We passes the value to the class and printing it.

Output:

```
$ g++ ptm4.cpp
$ a.out
Executed
```

9. Which is the best design choice for using pointer to member function?

- a) Interface
- b) Class
- c) Structure
- d) None of the mentioned

View Answer

Answer: a

Explanation: None.

10. What is the operation for .*?

- a) It combines the first operand and the second operand
- b) It separates the first operand and the second operand
- c) It reduces the data size
- d) None of the mentioned

View Answer

Answer: a

Explanation: The binary operator .* combines its first operand, which must be an object of class type, with its second operand, which must be a pointer-to-member type.

C++ Programming Questions and Answers – Free Store

1. Which is used to allocate and deallocate storage for objects during the execution?

- a) Stack
- b) Heap
- c) Freestore
- d) None of the mentioned

View Answer

Answer: c

Explanation: Free store is a pool of memory available for you to allocate and deallocate storage for objects during the execution of your program.

2. Which operators are used in the free store?

- a) new
- b) delete
- c) both new & delete
- d) none of the mentioned

View Answer

Answer: c

Explanation: new and delete operators are used to allocate and deallocate the memory for the program.

3. What type of class member is operator new?

- a) static
- b) dynamic
- c) const
- d) smart

View Answer

Answer: a

Explanation: None.

- a) A::operator delete
- b) B::operator delete
- c) Both A::operator delete & B::operator delete
- d) None of the mentioned

View Answer

Answer: b

Explanation: In this program, we are passing the value to the B, so we are printing B::operator delete.

Output:

```
$ g++ free.cpp
$ a.out
B::operator delete
```

- a) ~A()
- b) A :: operator delete[].
- c) B :: operator delete[].
- d) Warning

View Answer

Answer: d

Explanation: In this program, the behavior of the statement delete[] bp is undefined.

```
$ g++ a.cpp
a.cpp: In static member function 'static void A::operator delete [](void*, size_t)':
a.cpp:12: warning: deleting 'void*' is undefined
a.cpp: In static member function 'static void B::operator delete [](void*, size_t)':
a.cpp:20: warning: deleting 'void*' is undefined

$ a.out
~A()
~A()
~A()
A :: operator delete[].
```

- a) X::operator delete(void*)
 - b) Freeing 400 bytes
 - c) Depends on the compiler
 - d) Both X::operator delete(void*) & Depends on the compiler
- View Answer

Answer: d

Explanation: The memory value allocated for the program depends on compiler only.

```
$ g++ free2.cpp
$ a.out
X :: operator delete(void*)
Freeing 400 bytes
```

- a) X::operator new(size_t)
 - b) Error
 - c) Runtime error
 - d) None of the mentioned
- View Answer

Answer: c

Explanation: In this program, We are giving a location to two variables in the program, So it is arising an exception.

Output:

advertisement

```
$ g++ free3.cpp
$ a.out
X::operator new(size_t)
X::operator new(size_t, 0)
X::operator new(size_t, 1)
X::operator new(size_t, 2)
10000
10001
10002
X::operator new(size_t, 0)
Error: buffer location occupied
```

- a) Free store addr
 - b) Error
 - c) Segmentation fault
 - d) None of the mentioned
- View Answer

Answer: c

Explanation: In this program, The memory will go beyond the limit, So there will be exhaustion in memory.

Output:

```
$ g++ free4.cpp
$ a.out
```

free store addr = 0x80a8008
Segmentation fault

9. What must be an operand of operator delete?

- a) Pointer
- b) Array
- c) Stack
- d) None of the mentioned

View Answer

Answer: a

Explanation: The operand of delete must be a pointer returned by new.

10. How can object be allocated outside the object lifetime?

- a) int
- b) float
- c) void*
- d) none of the mentioned

View Answer

Answer: c

Explanation: None.

ayzom.com

C++ Programming Questions and Answers – Standard Library Design

1. Pick out the wrong header file about strings.

- a) <string>
- b) <regex>
- c) <ios>
- d) None of the mentioned

View Answer

Answer: c

Explanation: The standard header files for string is string and regex. So the wrong one presented here is ios.

2. Which is best for coding the standard library for c++?

- a) no trailing underscores on names
- b) complex objects are returned by value
- c) have a member-swap()
- d) all of the mentioned

View Answer

Answer: d

Explanation: None.

3. What is meant by vector in the container library contains?

- a) It is a sequence container that encapsulates dynamic size arrays
- b) It is a sequence container that encapsulates static size arrays
- c) It manages the memory
- d) None of the mentioned

View Answer

Answer: a

Explanation: vector in the container library contains sequence container that manipulates and encapsulates dynamic size arrays.

- a) 42
- b) 42 42
- c) 424
- d) 42 for 10 times

View Answer

Answer: d

Explanation: In this program, We used the vector to print the 42 for 10 times.

Output:

```
$ g++ std.cpp
$ a.out
42 42 42 42 42 42 42 42 42 42
```

- a) ab
- b) abc
- c) a
- d) error

View Answer

Answer: b

Explanation: We are using queue in this program and queue follows FIFO strategy to handle data hence the following output pattern is observed.

Output:

```
$ g++ std1.cpp
$ a.out
abc
```

- a) test
- b) test one
- c) test two
- d) none of the mentioned

View Answer

Answer: d

Explanation: In this program, We used the list to manipulate the given value.

Output:

```
$ g++ std3.cpp
$ a.out
test: one
two
test: one
two
test: three
three
three
test: three
three
```

7. Pick out the wrong header file.

- a) <algorithm>
- b) <containers>
- c) <iterators>
- d) <process>

View Answer

Answer: d

Explanation: There is no header file named <process> in C++.

8. What is meant by standard c++ library?

- a) It is the collection of class definitions for standard data structures and a collection of algorithms
- b) It is a header file
- c) Collection of algorithms
- d) None of the mentioned

View Answer

Answer: a

Explanation: It is the collection of class definitions for standard data structures. This part of the library was derived from the Standard Template Library.

9. Pick out parameter for rehash method in unordered_set in c++?

- a) count
- b) size
- c) hash
- d) none of the mentioned

View Answer

Answer: a

Explanation: count is used to return the new number of buckets.

10. What is the use of <exception> header

- a) Contains the standard exception files

- b) Contains the standard library files
- c) It is used to arise an exception in the program
- d) None of the mentioned

View Answer

Answer: a

Explanation: <exception> header file contains standard exception files used for exception handling in a C++ program.

ayzom.com

C++ Programming Questions and Answers – Container Design

1. How many sets of requirements are need in designing a container?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: c

Explanation: There are three sets of requirements. They are container interface requirements, Allocator interface requirements and iterator requirements.

2. Which interface in the container is required for storage management?

- a) Memory management
- b) Allocator interface
- c) Memory interface
- d) None of the mentioned

View Answer

Answer: b

Explanation: Allocator interface in the container is required for storage management.

3. Which is present in the basic interface of the allocator interface?

- a) Set of typedefs
- b) A pair of allocation functions
- c) allocate()
- d) All of the mentioned

View Answer

Answer: d

Explanation: The basic interface of an allocator class consists of a set of typedefs, a pair of allocation functions, allocate() and deallocate() and a pair of construction/destruction members, construct() and destroy().

- a) 345
- b) 678
- c) 901
- d) None of the mentioned

View Answer

Answer: d

Explanation: In this program, We are choosing and printing the numbers based on the certain limit and this is a composite design pattern.

Output:

```
$ g++ cont.cpp
$ a.out
0 1 2 3 4 5 6 7 8 9 10 11
3 4 5
6 7 8
9 10 11
```

5. Which container provides random access iterators?

- a) vector
- b) deque
- c) sort
- d) both vector & deque

View Answer

Answer: d

Explanation: Vector & deque container provides random access iterators.

- a) 2000
- b) No Space
- c) Error
- d) Depends on the compiler

View Answer

Answer: d

Explanation: In this program, We formed a simple container and got the size of it and printing it.

Output:

```
$ g++ cont1.cpp
$ a.out
200
```

7. Which is used for manually writing lookup table?

- a) std::map
- b) std::lookup
- c) std::find
- d) none of the mentioned

View Answer

Answer: a

Explanation: Lookup table means storing values in a table with unique keys for each value so that can be checked in future easily. So for such lookup tables maps are used in C++.

8. How can the member functions in the container be accessed?

- a) Iterator
- b) Indirect
- c) Both Iterator & Indirect
- d) None of the mentioned

View Answer

Answer: a

Explanation: The container manages the storage space for its elements and provides member functions to access them, either directly or through iterators which reference objects with similar properties to pointers.

9. Which of the following type does the container should define?

- a) Iterator type
- b) Vector type
- c) Storage type
- d) None of the mentioned

View Answer

Answer: a

Explanation: Every container must define an iterator type. Iterators allow algorithms to iterate over the container's contents.

10. Which are the parameters for the content of the buffer?

- a) start
- b) finish
- c) both start & finish
- d) none of the mentioned

View Answer

Answer: c

Explanation: The contents of the buffer are initialized using the values from the iterator range supplied to the constructor by the start and finish parameters.

ayzom.com

C++ Programming Questions and Answers – Vector

1. What do vectors represent?

- a) Static arrays
- b) Dynamic arrays
- c) Stack
- d) Queue

View Answer

Answer: b

Explanation: Vectors are sequence containers representing arrays that can change in size.

2. In which type of storage location are the vector members stored?

- a) Contiguous storage locations
- b) Non-contiguous storage locations
- c) Contiguous & Non-contiguous storage locations
- d) None of the mentioned

View Answer

Answer: a

Explanation: Vectors use contiguous storage locations for their elements, which means that their elements can also be accessed using offsets on regular pointers to its elements

3. How many vector container properties are there in c++?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: c

Explanation: There are three container properties in c++. They are sequence, Dynamic array and allocator-aware.

- a) 16
- b) 16 2
- c) 16 2 77
- d) 16 2 77 29

View Answer

Answer: d

Explanation: In this program, We got the values and printing it by using the vector and we are constructing vectors.

Output:

```
$ g++ vect.cpp
$ a.out
16 2 77 29
```

- a) 500
- b) 600
- c) 700
- d) Error

View Answer

Answer: b

Explanation: In this program, We are forming a stack and adding the elements and We are finding the total number of elements that are in stack.

Output:

```
$ g++ vect1.cpp
$ a.out
600
```

- a) Size of a 0
Size of b 3
b) Size of a 3
Size of b 5
c) Error
d) None of the mentioned
View Answer

Answer: a

Explanation: In this program, We are finding the size of the vector elements.

Output:

```
$ g++ vect2.cpp
$ a.out
Size of a 0
Size of b 3
```

- a) 10
b) 9
c) 8
d) 7
View Answer

Answer: d

Explanation: In this program, We are finding the size of the vector elements and resizing it.

Output:

advertisement

```
$ g++ vect3.cpp
$ a.out
7
```

- a) 10 20 0 100 0
b) 10 20 0 100
c) 10 20 0
d) 10 20
View Answer

Answer: a

Explanation: In this program, We are allocating the values to the vector and unallocated values are left as zero.

Output:

```
$ g++ vect4.cpp
$ a.out
10 20 0 100 0
```

9. Pick out the correct statement about vector.

- a) vector<int> values (5)
b) vector values (5)
c) vector<int> (5)
d) none of the mentioned
View Answer

Answer: a

Explanation: The syntax for declaring the vector element is `vector<type> variable_name (number_of_elements);`

10. Which is optional in the declaration of vector?

- a) Type
- b) Name
- c) Vector
- d) Number_of_elements

View Answer

Answer: d

Explanation: The number of elements is optional. An empty vector means, A vector that contains zero elements.

ayzom.com

C++ Programming Questions and Answers – Sequences

1. How many items are there in sequence container?

- a) 2
- b) 3
- c) 4
- d) 5

View Answer

Answer: d

Explanation: There are five items in sequence container. They are array, vector, list, forward_list and dequeue.

2. Which of the following class template are based on arrays?

- a) vector
- b) list
- c) dequeue
- d) both vector & dequeue

View Answer

Answer: d

Explanation: Class template vector and class template dequeue both are based on arrays.

3. Which of the following will return the new element at the end of container?

- a) front
- b) back
- c) push_back
- d) pop_back

View Answer

Answer: b

Explanation: Q3: back() in containers are used to access the last element of the sequence.

a) 12345

b) 1234

c) 54321

d) 43210

View Answer

Answer: c

Explanation: In this program, We used the operation of rbegin and rend on dequeue and produced the result.

Output:

```
$ g++ seq.cpp
$ a.out
5 4 3 2 1
```

a) a contains: 200 200 200 200 200b contains: 100 100 100

b) a contains: 100 100 100 100 100b contains: 200 200 200

c) a contains: 200 200 200 200 200b contains: 200 200 200

d) none of the mentioned

View Answer

Answer: a

Explanation: In this program, We swapped the values of both dequeues and printing the dequeues.

Output:

```
$ g++ seq1.cpp
$ a.out
a contains: 200 200 200 200 200b contains: 100 100 100
```

- a) 110
- b) 220
- c) Both 110 & 220
- d) None of the mentioned

View Answer

Answer: c

Explanation: In this program, We cleared the old values presented in the dequeue with the new values.

Output:

```
$ g++ seq2.cpp
$ a.out
110 220
```

- a) 1 2 3 4 5
- b) 0 1 2 3 4
- c) 1 2 3 4
- d) 5 4 3 2 1

View Answer

Answer: b

Explanation: In this program, We allocated the values to the vector by using get allocator and then we are destroying it.

Output:

```
$ g++ seq3.cpp
$ a.out
0 1 2 3 4
```

- a) 2.72 12.15 72.25
- b) 12.15 73.0 12.77
- c) 73.35
- d) None of the mentioned

View Answer

Answer: a

Explanation: In this program, We are eliminating the values by using the unique operation in the list.

Output:

```
$ g++ seq4.cpp
$ a.out
2.72 12.15 72.25
```

9. How the list containers are implemented?

- a) Using Double linked list
- b) Using Single linked list
- c) Using Single & Double linked list
- d) None of the mentioned

View Answer

Answer: a

Explanation: List containers are implemented as doubly-linked lists. Doubly linked lists can store each of the elements they contain in different and unrelated storage locations.

10. Which of the following does not support any insertion or deletion?

- a) Array
- b) Vector
- c) Dequeue
- d) List

[View Answer](#)

Answer: a

Explanation: Because array is not dynamic in nature, So they can't be manipulated.

ayzom.com

C++ Programming Questions and Answers – Sequence Adapters

1. What do container adapter provide to interface?

- a) Restricted interface
- b) More interface
- c) No interface
- d) None of the mentioned

View Answer

Answer: a

Explanation: A container adapter provides a restricted interface to a container. In particular, adapters do not provide iterators; they are intended to be used only through their specialized interfaces.

2. What does the sequence adaptor provide?

- a) Insertion
- b) Deletion
- c) Interface to sequence container
- d) None of the mentioned

View Answer

Answer: c

Explanation: Sequence adaptor provides interface to sequence container.

3. Which are presented in the container adaptors?

- a) stack
- b) queue
- c) priority_queue
- d) all of the mentioned

View Answer

Answer: d

Explanation: These mentioned things are presented in container adapters.

- a) 12
- b) 75
- c) 63
- d) None of the mentioned

View Answer

Answer: c

Explanation: In this program, We used the queue operation and performed the back operation. Because of that operation, We got the output as 63.

Output:

```
$ g++ sca.cpp
$ a.out
63
```

- a) 51
- b) 52
- c) 54
- d) 55

View Answer

Answer: d

Explanation: In this program, We used the push and pop operation of queue to find out the total of all the number from 1 to 10.

Output:

```
$ g++ sca1.cpp
$ a.out
55
```

- a) 100 40 30 25
- b) 100 40 30
- c) 100 40
- d) None of the mentioned

View Answer

Answer: a

Explanation: In this program, We used priority_queue and with that we are pushing and popping out the elements.

Output:

```
$ g++ sca2.cpp
$ a.out
100 40 30 25
```

- a) 05
- b) 15
- c) 24
- d) 102

View Answer

Answer: a

Explanation: In this program, We declared myints and not initialized in first option, So it's value is 0 and on another, We are pushing 5 values, So it's size is 5.

Output:

advertisement

```
$ g++ sca3.cpp
$ a.out
05
```

- a) 10
- b) 20
- c) 13
- d) 15

View Answer

Answer: d

Explanation: In this program, We used top option and this will return the reference to the next element.

Output:

```
$ g++ sca4.cpp
$ a.out
15
```

9. In which context does the stack operates?

- a) FIFO
- b) LIFO
- c) Both FIFO & LIFO
- d) None of the mentioned

View Answer

Answer: b

Explanation: A stack is a container where elements operate in a LIFO context, where elements are inserted (pushed) and removed (popped) from the end of the container.

10. Which operator is used in priority queue?

- a) operator<
- b) operator>
- c) operator)
- d) none of the mentioned

View Answer

Answer: a

Explanation: It is used to decide the priority of two elements to be inserted in the queue.

ayzom.com

C++ Programming Questions and Answers – Associative Containers

1. What do associate containers implement?

- a) Arrays
- b) Associative arrays
- c) Functional Arrays
- d) None of the mentioned

View Answer

Answer: b

Explanation: Associative containers refer to a group of class templates in the standard library of the C++ programming language that implement ordered associative arrays.

2. By using which of the following the elements in the associate container can be efficiently accessed?

- a) Key
- b) Position
- c) Both Key & Position
- d) None of the mentioned

View Answer

Answer: a

Explanation: Associative containers are designed to be especially efficient in accessing its elements by their key, as opposed to sequence containers which are more efficient in accessing elements by their position.

3. How many items are presented in the associate container?

- a) 2
- b) 3
- c) 4
- d) 5

View Answer

Answer: c

Explanation: There are 4 items presented in the associate container. They are set, multiset, map and multimap.

a) 0000

b) 0001

c) 0011

d) 1111

View Answer

Answer: d

Explanation: In this program, We converted the bitset values to string and printing it.

Output:

```
$ g++ asc.cpp
$ a.out
1111
```

a) first contains: 10 33 33 33 10

second contains: 10 10 10 33 33

b) first contains: 10 33 33 33 10

second contains: 10 10 10 33 10

c) first contains: 10 33 33 33 30

second contains: 10 10 10 33 10

d) none of the mentioned

View Answer

Answer: a

Explanation: In this program, We swapped the values according to their position.

Output:

```
$ g++ asc1.cpp
$ a.out
first contains: 10 33 33 33 10
second contains: 10 10 10 33 33
```

- a) a => 200
- c => 300
- b) a => 200
- b => 100
- c) a => 200
- b => 100
- c => 300
- d) None of the mentioned

View Answer

Answer: c

Explanation: In this program, We used the map template and the we used the begin operation and then we are printing the elements.

Output:

```
$ g++ asc2.cpp
$ a.out
a => 200
b => 100
c => 300
```

- a) 10
- b) 20
- c) 30
- d) All of the mentioned

View Answer

Answer: d

Explanation: In this program, We used the set template and then we are initializing the values and then we are erasing it.

Output:

advertisement

```
$ g++ asc3.cpp
$ a.out
10 20 30
```

- a) 12345
- b) 01234
- c) 1234
- d) 0123

View Answer

Answer: b

Explanation: In this program, We used the set template and then we compared the keys and printing the result.

Output:

```
$ g++ asc4.cpp
$ a.out
0 1 2 3 4
```

9. How many instance are allowed by map and set while inserting a element into container?

- a) 1
- b) 2
- c) 3
- d) Multiple

[View Answer](#)

Answer: a

Explanation: Both map and set only allow one instance of a key or element to be inserted into the container.

10. What do maps and sets support?

- a) Single directional iterators
- b) Bi-directional iterators
- c) Single & Bi-directional directional iterators
- d) None of the mentioned

[View Answer](#)

Answer: b

Explanation: Bi-directional iterator are used to move in both direction from any element i.e. both forward and backward movement are allowed.

ayzom.com

C++ Programming Questions and Answers – Almost Containers

1. What kind of library is Standard Template Library?

- a) Polymorphic
- b) Generic
- c) Both Polymorphic & Generic
- d) None of the mentioned

View Answer

Answer: b

Explanation: The STL is a generic library, meaning that its components are heavily parameterized.

2. To what type of object does the container can be instantiated?

- a) int
- b) float
- c) double
- d) any type of object

View Answer

Answer: d

Explanation: All type of object does the container can be instantiated.

3. What type of class template is list?

- a) Class-based
- b) Node-based
- c) Method-based
- d) None of the mentioned

View Answer

Answer: b

Explanation: It is node-based because it allows efficient insertion anywhere on the program.

4. What type of access does deque and vector provide?

- a) Linear access
- b) Parallel access
- c) Random access
- d) None of the mentioned

View Answer

Answer: c

Explanation: Because they can manipulate the values on anywhere in the program, So it is providing random access.

5. Where does the vector add the item?

- a) End
- b) Insert
- c) Middle
- d) None of the mentioned

View Answer

Answer: a

Explanation: Vector allows insertion of element at the end.

6. Which are not full container classes in c++?

- a) Sequence container
- b) Associative container

- c) Container adaptor
 - d) None of the mentioned
- View Answer

Answer: c

Explanation: Container adaptors are not full container classes, but classes that provide a specific interface relying on an object of one of the container classes such as deque or list to handle the elements.

7. What is the lifetime of the element in container?

- a) Whole program
- b) Outside the block
- c) Everywhere
- d) Only on that container

View Answer

Answer: d

Explanation: A Container “owns” its elements: the lifetime of an element stored in a container cannot exceed that of the Container itself.

- a) 1
- b) 2
- c) 4
- d) 3

View Answer

Answer: c

Explanation: In this program, We are counting the number of elements in the map.

Output:

```
$ g++ alc.cpp
$ a.out
4
```

- a) 15
- b) 20
- c) 10
- d) Error

View Answer

Answer: b

Explanation: In this program, We used the queue template and the top method is used to retain the last but before element.

Output:

```
$ g++ alc1.cpp
$ a.out
20
```

- a) y=> 202
- b) y=> 252
- c) y=> 202 & y=> 252
- d) None of the mentioned

View Answer

Answer: a

Explanation: In this program, the method rbegin is used to return the first element in the map.

Output:

```
$ g++ alc2.cpp
$ a.out
y = > 202
```

ayzom.com

C++ Programming Questions and Answers – Defining a New Container

1. What do all STL containers define?

- a) Iterator types
- b) Begin methods
- c) End methods
- d) All of the mentioned

View Answer

Answer: d

Explanation: All the STL containers define the iterator types for that container, e.g., iterator and const_iterator, e.g., vector::iterator and the begin/end methods for that container, e.g., begin() and end().

2. What do we return if we use simple array on a internal container?

- a) Methods
- b) Pointers
- c) Objects
- d) None of the mentioned

View Answer

Answer: b

Explanation: Pointers are legal iterators, so if your internal container is a simple C array, then all you need to do is return the pointers.

3. What is mandatory for designing a new container?

- a) Classes
- b) Iterators
- c) Container
- d) None of the mentioned

View Answer

Answer: b

Explanation: Iterators are used to increase the generality of an algorithm. Otherwise we need to define the algorithm for each types.

4. What are the design requirements for building a container from the scratch?

- a) Container interface requirements
- b) Allocator interface requirements
- c) Iterator requirements
- d) All of the mentioned

View Answer

Answer: d

Explanation: These are the design specific requirements for building a container from the scratch.

5. How many iterators are needed for the defining a new container?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: c

Explanation: There are three main iterators needed for designing a container. They are const iterator, Reverse iterator and Iterator traits.

6. What is the use of the allocator interface in the user-defined container?

- a) Storage management

- b) Memory management
 - c) Storage & Memory management
 - d) None of the mentioned
- View Answer

Answer: a

Explanation: None.

7. How many types of container classes are there in c++?

- a) 1
- b) 2
- c) 3
- d) As many as possible

View Answer

Answer: b

Explanation: There are two type of container classes in c++. They are value containers and reference containers.

8. What is the name of the container which contains group of multiple objects?

- a) Heterogeneous container
- b) Homogeneous container
- c) Both Homogeneous & Heterogeneous container
- d) None of the mentioned

View Answer

Answer: a

Explanation: None.

- a) spaces
- b) spaces in
- c) spaces in text
- d) spacesintext

View Answer

Answer: d

Explanation: In this program, We formed a algorithm to remove spaces in the string.

Output:

```
$ g++ dan.cpp
$ a.out
spacesintext
```

- a) 0
- b) 1
- c) 2
- d) 0 1 4

View Answer

Answer: d

Explanation: In this program, We formed an algorithm to find the square of the given number.

Output:

```
$ g++ dan1.cpp
$ a.out
0 1 4
```

C++ Programming Questions and Answers – seq_con Array Class – 1

1. What is sequence container arrays?

- a) C-like arrays
- b) Template class sequence container, alternative for C-like arrays
- c) Collection of data of the same type
- d) Collection of objects

View Answer

Answer: b

Explanation: Sequence Containers arrays are an alternative for C-like arrays. It is a static continuous array that uses template classes with extended features for array implementation.

2. Pick the correct statement.

- a) Sequence Container arrays know (somehow stores within) its size whereas C-like arrays do not
- b) Sequence Container arrays have no advantage over C-like arrays
- c) Sequence Container arrays are same as C-like arrays
- d) Sequence Container arrays are also present in C

View Answer

Answer: a

Explanation: Sequence Containers Arrays stores its size within itself so need to pass extra size parameter when passing this array as an argument.

3. Which of the following is/are advantage(s) of Sequence Container arrays over C-like arrays?

- a) Sequence Container arrays store its size within itself whereas C-like arrays do not
- b) Sequence Container arrays are more efficient
- c) Sequence Container arrays have no array decay problem whereas C-like arrays do have
- d) All of the mentioned

View Answer

Answer: d

Explanation: Sequence Container arrays(a.k.a Array classes) somehow stores its size and it can be implemented efficiently. Also, Array classes do not have Array decay problem.

4. Which of the following function(s) of Array classes are similar to [] operator?

- a) at()
- b) get()
- c) both at() and get()
- d) front()

View Answer

Answer: c

Explanation: Both at() and get() function are used to access the elements stored at ith position of the array.

5. How many different ways are there to access an element of array classes at the ith position?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: c

Explanation: There are three ways of accessing Array classes as mentioned below:

i. using [] operator(same as C-like arrays)

- ii. using at() function available in array classes.
- iii. using get() function not a member of the array class.

6. What header file is included to use array classes?

- a) <array>
- b) <Array>
- c) <algorithm>
- d) <ARRAY>

View Answer

Answer: a

Explanation: <array> header file is provided by the C++ to use array classes.

7. What is the correct syntax of declaring an array class?

- a) array<type> arr;
- b) array<type,size> arr;
- c) Array<type> arr;
- d) Array<type,size> arr;

View Answer

Answer: b

Explanation: The declaration of array class starts with a keyword array followed by <> specifying the type and size of array and then the name of the identifier. Example: array<int, 10> arr; arr is an array class of type in with size = 10.

d) Printing Using at() function: 1 2 3 4 5

View Answer

Answer: b

Explanation: In this program we are trying to print the array first using [] operator then using the at() function of the array class.

Output:

```
$ ./a.out
Printing Using [] operator: 1 2 3 4 5
Printing Using at() function: 1 2 3 4 5
```

9. What is the syntax of printing the first element of an array Arr using get() function?

- a) Arr.get(0)
- b) get<0>(Arr)
- c) Arr.get[0]
- d) get<0>[Arr]

View Answer

Answer: b

Explanation: To access the first element of an array class Arr using get() function, we use the following get<index>(Arr) where index is an integer constant number, not an identifier.

10. Which header file is required to use get() function?

- a) <array>
- b) <tuple>
- c) <Array>
- d) <access>

View Answer

Answer: b

Explanation: <tuple> header file is required to use the get() function for accessing an element.

11. What is the difference between get() and at()?

- a) at() is available under <array> header file whereas get() is available under <tuple> header file
- b) at() is a member function of array class whereas get() is not
- c) get() takes array class as a parameter whereas at() takes a constant integer(i.e. index) as a parameter
- d) all of the mentioned

View Answer

Answer: d

Explanation: get() and at() differ in various ways. get() is not a part of array class, get is available under <tuple> header and get() takes array class also as a parameter to access the element.

12. Which function is used to access the first element of an array class?

- a) front()
- b) start()
- c) back()
- d) first()

View Answer

Answer: a

Explanation: Array class provides front() function to access the first element of the array class.

13. Which function is used to access the last element of an array class?

- a) end()
- b) start()
- c) back()
- d) last()

View Answer

Answer: c

Explanation: Array class provides back() function to access the last element of the array class.

14. Which of the following function(s) is/are used to get the size of the array class?

- a) size()
- b) max_size()
- c) both size() and max_size()
- d) get_size()

View Answer

Answer: c

Explanation: Both size() and max_size() are used to get the size of array class. There is no difference between size() and max_size() of array class.

View Answer

Answer: a

Explanation: Both size() and max_size() returns the same value i.e. the size of array defined during declaration. Therefore both prints the value 10.

Output:

```
$ ./a.out
size:10
maxsize:10
```

C++ Programming Questions and Answers – seq_con Array Class – 2

1. What is the use of swap() function in array class?

- a) Swaps two elements of an array given elements
- b) Swaps two arrays
- c) Swaps two elements given indices of elements
- d) Swaps same elements of the array if required

View Answer

Answer: b

Explanation: swap() function is used to swap elements of two array classes provided the size of both arrays classes are same.

2. What is the syntax of swap()?

- a) swap(arr1, arr2);
- b) arr1.swap(arr2);
- c) swap<int, int>(arr1, arr2);
- d) swap[arr1, arr2];

View Answer

Answer: b

Explanation: The correct syntax of swap function is arr1.swap(arr2) i.e. one array calling swap() function with second array as parameter to swap function. Also swap is a function therefore [] operator cannot be used to call swap function.

View Answer

Answer: a

Explanation: arr1 has elements from 1-5 and arr2 has elements 6-10 initially. After swapping arr1 has elements from 6-10 and arr2 has elements from 1-5. Therefore output is 6 7 8 9 10 then 1 2 3 4 5.

Output:

```
$ ./a.out
6 7 8 9 10
1 2 3 4 5
```

- c) Error
- d) Segmentation fault

View Answer

Answer: c

Explanation: As the size of both the array classes is not equal therefore the swap function gives an error stating that no matching function available.

5. What is the use of empty() function in array classes?

- a) To check whether the size of an array is zero or not
- b) To check whether an array is empty or not
- c) To check how many elements are there in the array
- d) To check whether an array contains negative elements or not

View Answer

Answer: a

Explanation: empty() function is used to check whether the size of an array class is zero or not. It is not used to check whether an array is empty or not. The function true only if size/max_size of an array is zero otherwise it returns false.

6. What is the use of fill() function in array class?

- a) To fill an array with a given single value
- b) To delete all the elements that are equal to the given value
- c) To replace all the elements of the array which are equal to the given value

d) To check whether given element fills the array or not

View Answer

Answer: a

Explanation: fill() function is used to fill an array class with the given single value.

a) 22222

b) 20000

c) 00002

d) 20002

View Answer

Answer: a

Explanation: fill() function sets the value of each element equal to the value passed as parameter to the function.

c) Error

d) Segmentation fault

View Answer

Answer: c

Explanation: swap() function is used for swapping two array classes not two C-like arrays. Therefore the swap() function gives error.

a) 5

b) Compile-time error

c) Run-time error

d) Segmentation fault

View Answer

Answer: b

Explanation: The compiler detects that the array class size is 5 and we are trying to access the 5th index which is out of bound therefore the program gives error.

a) Program 1 gives compile-time error and Program 2 gives run-time error

b) Program 1 gives run-time error and Program 2 gives compile-time error

c) Both programs results into compile-time error

d) Both programs results into run-time error

View Answer

Answer: a

Explanation: The Program 1 gives compile-time error whereas Program 2 gives run-time error. This is because get() function takes constant integer as the argument for accessing element of the array, therefore at compile time only the compiler verifies whether the index is accessible or not as we know the array class size during compile time, Whereas in case of at() function it takes variable as the parameter for accessing element, therefore the index range is checked during run-time therefore the error is detected during run-time.

C++ Programming Questions and Answers – seq_con Vector Class – 1

1. What are the vectors?

- a) Arrays with dynamic size
- b) Arrays with different types of elements
- c) Same as array classes
- d) Arrays with static size but use template classes

View Answer

Answer: a

Explanation: Vectors are just like arrays with the ability to resize itself whenever an element is added to or deleted from it.

2. Pick the correct statement.

- a) Vectors have dynamic size whereas Array classes have a static size
- b) Both vectors and Array classes have a dynamic size
- c) Both vectors and Array classes have a static size
- d) Vectors have static size whereas Array classes have a dynamic size

View Answer

Answer: a

Explanation: Vectors are implemented in a way so that it can handle any number of elements at a time means the size of a vector can vary, whereas Array classes have fixed size.

3. Pick the incorrect statement.

- a) Vectors have a dynamic size
- b) Vectors are placed in contiguous storage
- c) Insertion in vectors always takes constant time
- d) All of the mentioned

View Answer

Answer: c

Explanation: Insertion in vectors are not always constant. When we are inserting an element at the end of the vector then if a vector is full then it needs to size itself which takes time to resize and time to insert element else just time for inserting that element at the end. Hence the insertion time is not constant always. Vectors have a dynamic size. They are placed in contiguous memory for easy access.

4. Which of the following header file is needed to use vectors in your program?

- a) <array>
- b) <vector>
- c) <containers>
- d) <stdio>

View Answer

Answer: b

Explanation: Header file <vector> contains all the implementation of vector methods, hence we need to include this header file.

5. Which of the following(s) can be used to access the first element of a vector v?

- a) v.begin()
- b) v.cbegin()
- c) v[0]
- d) all of the mentioned

View Answer

Answer: d

Explanation: To access the first element of a vector we can use the following things:

- i) v.begin()

- ii) `v.cbegin()`
- iii) `v[0]`
- iv) `v.at(0)`

6. Which of the following(s) can be used to access the last element of a vector `v`?

- a) `v.end()`
- b) `v.cend()`
- c) both `v.end()` and `v.cend()`
- d) vectors do not have a function to access the last element

View Answer

Answer: d

Explanation: There are no function to access the last element of the vector. The `end()` and `cend()` returns the iterator to an element which is kept at the last of the vector to keep the knowledge about the end of a vector. In order to access the last element, you can first find the size and then can use `v[size-1]` or `v.at(size - 1)` to access the last element.

7. What is the difference between `begin()` and `cbegin()` in vectors?

- a) both are same
- b) `begin()` returns iterator to first element and `cbegin()` returns iterator to last element
- c) `begin()` returns an iterator to first element whereas `cbegin()` returns constant iterator to first element
- d) `begin()` returns first element `cbegin()` returns void

View Answer

Answer: c

Explanation: Both `begin()` and `cbegin()` are used to access the first element of the vector. The function `begin()` returns an iterator to first element whereas `cbegin()` returns a constant iterator to first element.

8. What is the difference between `begin()` and `rbegin()`?

- a) both are the same
- b) `begin()` returns an iterator to the first element and `rbegin()` returns an iterator to an element kept at the end of the vector
- c) `begin()` returns an iterator to first element whereas `rbegin()` returns constant iterator to first element
- d) `begin()` returns first element `rbegin()` returns void

View Answer

Answer: b

Explanation: `begin()` is used to return the iterator to the first element of the vector whereas `rbegin()` is used to return the an element stored at in the last of a vector.

9. Which is the following is syntactically correct for vector `v`?

- a) `vector<int> :: const_iterator itr = v.rbegin();`
- b) `vector<int> :: reverse_iterator itr = v.begin();`
- c) `vector<int> :: iterator itr = v.begin();`
- d) `vector<int> :: iterator itr = v.cbegin();`

View Answer

Answer: c

Explanation: `v.rbegin()` returns iterator of reverse iterator therefore cannot be stored in `const_iterator` (type mismatch). Similarly `v.begin()` returns normal iterator therefore cannot be stored in `reverse_iterator` and `v.cbegin()` returns the `const_iterator` therefore cannot be stored in normal iterator.

- a) 1 2 3 4 5
- b) 1 3 5
- c) 1 4 5
- d) Error

View Answer

Answer: a

Explanation: A normal iterator can be stored in const_iterator therefore program does not gives any error hence will be executed perfectly.
Output:

```
$ ./a.out  
1 2 3 4 5
```

- a) 1 2 3 4 5
- b) 3 2 3 4 5
- c) 5 4 3 2 1
- d) 3 3 3 3 3

View Answer

Answer: b

Explanation: We have changed the value of 0th element of vector from 1 to 3 therefore the output is as follows.

Output:

```
$ ./a.out  
3 2 3 4 5
```

- a) 1 2 3 4 5
- b) 3 2 3 4 5
- c) Error
- d) Segmentation fault

View Answer

Answer: c

Explanation: As i is a constant iterator therefore value stored in it is read-only therefore cannot be updated. Therefore the program gives an error.

13. Which of the following function is used to get the actual number of elements stored in vector?

- a) v.size()
- b) v.capacity()
- c) v.max_size()
- d) v.no_of_elements()

View Answer

Answer: a

Explanation: To get the number of elements stored in the vector v we use the function v.size(). It returns how many elements are currently in the vector excluding the void places.

14. Which function is used to get the total capacity of a vector?

- a) v.size()
- b) v.capacity()
- c) v.max_size()
- d) v.no_of_elements()

View Answer

Answer: b

Explanation: capacity() function is used to get the total number of elements that can be stored at present in the vector.

15. How the size of a vector increases once it is full?

- a) Vector increases its capacity one by one
- b) Vector doubles its capacity after it is full
- c) Vector increases its capacity by half of its previous size
- d) Vector increases its capacity by a constant factor

View Answer

Answer: b

Explanation: Once the vector is full i.e. number of elements in the vector becomes equal to the capacity of the vector then vector doubles its capacity i.e. if previous capacity was 2 then new capacity becomes $2 * 2 = 4$ or $2 + 2 = 4$.

ayzom.com

C++ Programming Questions and Answers – seq_con Vector Class – 2

3. What is the output of the following C++ code?

```
#include <iostream>
#include <vector>

using namespace std;

int main()
{
    vector<int> v;

    for (int i = 1; i <= 5; i++)
        v.push_back(i);
    v.resize(4);
    for (auto it = v.begin(); it != v.end(); it++)
        cout << *it << " ";
    return 0;
}
```

View Answer

Answer: b

Explanation: The size() returns the number of elements in the vector and capacity() returns the total number of elements that this vector can hold. Hence as the number of elements in vector is 5 and size is increased by 2 times. Therefore output is 5 and 8

Output:

```
$ ./a.out
5
8
```

3. What is the output of the following C++ code?

```
#include <iostream>
#include <vector>

using namespace std;

int main()
{
    vector<int> v;

    for (int i = 1; i <= 5; i++)
        v.push_back(i);
    v.resize(4);
    for (auto it = v.begin(); it != v.end(); it++)
        cout << *it << " ";
    return 0;
}
```

2. Which function is used to check whether the vector is empty or not?

- a) empty()
- b) isempty()
- c) haveElements()
- d) none()

View Answer

Answer: a

Explanation: empty() function is provided by the vector container to check whether it is empty or not.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <vector>
```

```
using namespace std;

int main()
{
    vector<int> v;

    for (int i = 1; i <= 5; i++)
        v.push_back(i);
    v.resize(4);
    for (auto it = v.begin(); it != v.end(); it++)
        cout << *it << " ";
    return 0;
}
```

- a) 1 2 3 4 5
- b) 1 2 3 4
- c) 2 3 4 5
- d) error

View Answer

Answer: b

Explanation: `resize()` function is used to resize a vector container. It updates the size of vector and removes all the elements after `n` if new size(`n`) is less than previous size. Hence in the program initially the vector has 5 elements but after resizing the vector to 4 it has only 4 elements as 5 is removed.

Output:

```
$ ./a.out
1 2 3 4
```

3. What is the output of the following C++ code?

```
#include <iostream>
#include <vector>

using namespace std;

int main()
{
    vector<int> v;

    for (int i = 1; i <= 5; i++)
        v.push_back(i);
    v.resize(4);
    for (auto it = v.begin(); it != v.end(); it++)
        cout << *it << " ";
    return 0;
}
```

- d) Error

View Answer

Answer: a

Explanation: Initially the size of the vector is 5 as it contains only 5 elements. After resizing the elements 5 is terminated so only 4 remains therefore the size becomes 4. Hence output is as follow.

Output:

```
$ ./a.out
5
4
```

3. What is the output of the following C++ code?

```
#include <iostream>
#include <vector>
```

```
using namespace std;

int main()
{
    vector<int> v;

    for (int i = 1; i <= 5; i++)
        v.push_back(i);
    v.resize(4);
    for (auto it = v.begin(); it != v.end(); it++)
        cout << *it << " ";
    return 0;
}
```

View Answer

Answer: b

Explanation: The capacity denotes how many elements a vector can hold. On resizing a vector the capacity of a vector is not changed hence the capacity before and after is same. Therefore the output is as follows.

Output:

```
$ ./a.out
8
8
```

3. What is the output of the following C++ code?

```
#include <iostream>
#include <vector>

using namespace std;

int main()
{
    vector<int> v;

    for (int i = 1; i <= 5; i++)
        v.push_back(i);
    v.resize(4);
    for (auto it = v.begin(); it != v.end(); it++)
        cout << *it << " ";
    return 0;
}
```

c) Error

d) Segmentation fault

View Answer

Answer: b

Explanation: Initially we have 5 elements in the vector therefore the capacity of the vector is 8 (one can observe that as capacity doubles after vector is full). Now the function `shrink_to_fit()` makes the capacity of vector equal to its size hence removing the extra space occupied by the vector. Therefore as only 5 elements were there in the vector therefore the capacity becomes 8.

Output:

```
$ ./a.out
8
5
```

3. What is the output of the following C++ code?

```
#include <iostream>
#include <vector>

using namespace std;

int main()
```



```

{
    vector<int> v;

    for (int i = 1; i <= 5; i++)
        v.push_back(i);
    v.resize(4);
    for (auto it = v.begin(); it != v.end(); it++)
        cout << *it << " ";
    return 0;
}

```

- a) 8
- b) 10
- c) 5
- d) 6

View Answer

Answer: b

Explanation: After shrinking the capacity of vector the capacity of vector becomes 5. Now when a new element i.e. 10 is inserted into the vector then the capacity of the vector will double i.e. it will become 10. hence the final capacity will be 10.

3. What is the output of the following C++ code?

```

#include <iostream>
#include <vector>

using namespace std;

int main()
{
    vector<int> v;

    for (int i = 1; i <= 5; i++)
        v.push_back(i);
    v.resize(4);
    for (auto it = v.begin(); it != v.end(); it++)
        cout << *it << " ";
    return 0;
}

```

- a) 10
- b) 8
- c) 50
- d) 60

View Answer

Answer: c

Explanation: In this program reserve(n) function is used which is used to reserve the space for n elements in vector. Hence when the reserve(50) function is called for vector v then the we are trying to reserve memory for 50 elements, hence the capacity of vector v becomes 50.

3. What is the output of the following C++ code?

```

#include <iostream>
#include <vector>

using namespace std;

int main()
{
    vector<int> v;

    for (int i = 1; i <= 5; i++)
        v.push_back(i);
    v.resize(4);
    for (auto it = v.begin(); it != v.end(); it++)
        cout << *it << " ";
}

```

```
    return 0;
}
```

- a) 5
- b) 1
- c) 4
- d) 3

View Answer

Answer: c

Explanation: data() function in vector returns the direct pointer to the memory array which the vector has used to store its elements. Hence a pointer to vector is returned. So when we are accessing *(pos + 3) we are trying to do v[3] which is 4. Hence the output is as follows.

Output:

```
$ ./a.out
4
```

3. What is the output of the following C++ code?

```
#include <iostream>
#include <vector>

using namespace std;

int main()
{
    vector<int> v;

    for (int i = 1; i <= 5; i++)
        v.push_back(i);
    v.resize(4);
    for (auto it = v.begin(); it != v.end(); it++)
        cout << *it << " ";
    return 0;
}
```

10. Which of the following function is used to insert an element at the end of a vector?

- a) push_back()
- b) pop_back()
- c) front()
- d) end()

View Answer

Answer: a

Explanation: Vector provides push_back() function to insert an element at its end.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <vector>

using namespace std;

int main()
{
    vector<int> v;

    for (int i = 1; i <= 5; i++)
        v.push_back(i);
    v.resize(4);
    for (auto it = v.begin(); it != v.end(); it++)
        cout << *it << " ";
    return 0;
}
```

View Answer

Answer: c

Explanation: Vector never reduces its capacity on deleting any element which one may get confuse by thinking about the fact that vector doubles its memory on insertion. Hence both returns the same capacity.

Output:

```
$ ./a.out
8
8
```

3. What is the output of the following C++ code?

```
#include <iostream>
#include <vector>

using namespace std;

int main()
{
    vector<int> v;

    for (int i = 1; i <= 5; i++)
        v.push_back(i);
    v.resize(4);
    for (auto it = v.begin(); it != v.end(); it++)
        cout << *it << " ";
    return 0;
}
```

d) 1 2 3 4 5

View Answer

Answer: a

Explanation: assign(m,n) function changes the vector values by assigning new values to vector. It copies m times the value n to the vector by first removing all the initial values. hence the vector has 3 8's after updation i.e. using assign(3,8) function. Hence the output is as follows.

Output:

```
$ ./a.out
1 2 3 4 5
8 8 8
```

3. What is the output of the following C++ code?

```
#include <iostream>
#include <vector>

using namespace std;

int main()
{
    vector<int> v;

    for (int i = 1; i <= 5; i++)
        v.push_back(i);
    v.resize(4);
    for (auto it = v.begin(); it != v.end(); it++)
        cout << *it << " ";
    return 0;
}
```

13. Which function is used to swap two vectors?

- a) swap()
- b) change()
- c) merge()
- d) exchange()

View Answer

Answer: a

Explanation: Vectors allows the use of swap function to swap to vectors with each other of same type and size.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <vector>

using namespace std;

int main()
{
    vector<int> v;

    for (int i = 1; i <= 5; i++)
        v.push_back(i);
    v.resize(4);
    for (auto it = v.begin(); it != v.end(); it++)
        cout << *it << " ";
    return 0;
}
```

- a) 1 2 3 4 5 6 7 8 9 10
- b) 6 7 8 9 10 1 2 3 4 5
- c) Error
- d) Segmentation fault

View Answer

Answer: c

Explanation: swap() function in vector allows to swap two vectors of same size and type but here the vectors v1 and v2 have different types therefore the program gives the error.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <vector>

using namespace std;

int main()
{
    vector<int> v;

    for (int i = 1; i <= 5; i++)
        v.push_back(i);
    v.resize(4);
    for (auto it = v.begin(); it != v.end(); it++)
        cout << *it << " ";
    return 0;
}
```

- a) 1 2 3 4 5 6 7 8 9 10
- b) 6 7 8 9 10 1 2 3 4 5
- c) Error
- d) Segmentation fault

View Answer

Answer: b

Explanation: Here swap() function is used and the type and size of both vectors v1 and v2 are same therefore they can be swapped and hence the program allows such swap. Therefore no error and program runs perfectly.

Output:

```
$ ./a.out
```

ayzom.com

C++ Programming Questions and Answers – seq_con List

1. How many list sequence containers are provided by STL?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: b

Explanation: There are two list sequence containers are provided by STL namely forward_list and list.

2. Which type of list a Forward_list sequence container implements?

- a) Singly Linked List
- b) Doubly Linked List
- c) Both type of list
- d) A simple sequence of array

View Answer

Answer: a

Explanation: Forward_list sequence container implements a Singly Linked List.

3. Which type of list a List sequence container implements?

- a) Singly Linked List
- b) Doubly Linked List
- c) Both type of list
- d) A simple sequence of array

View Answer

Answer: b

Explanation: List sequence container implements Doubly Linked List.

4. Which of the following header file is required for forward_list?

- a) <forward_list>
- b) <list>
- c) <f_list>
- d) <Forward_List>

View Answer

Answer: a

Explanation: One needs to implement <forward_list> header file to use forward_list in a program.

5. Which of the following(s) is/are the correct way of assigning values to a forward_list f?

- a) f.assign({1,2,3,4,5})
- b) f.assign(10,5)
- c) both f.assign({1,2,3,4,5}) and f.assign(10,5)
- d) f.assign(1,1,1,1)

View Answer

Answer: c

Explanation: Both f.assign({1,2,3,4,5}) and f.assign(10,5) are correct way of assigning values to a forward_list. The first assignment initializes the list with the elements 1,2,3,4 and 5 whereas the second assignment initializes the list 10 elements with value 5 i.e. 5 10 times.

6. How the list differs from vectors?

- a) Vector is contiguous whereas List is non-contiguous

- b) Insertion in the list takes constant time whereas it is not constant in vectors
- c) There is no capacity defined for list
- d) All of the mentioned

View Answer

Answer: d

Explanation: List is non-contiguous that means elements of a list are not the contiguous manner in memory. Insertion in a list is constant for because we are not increasing the size of the list anywhere which was the case of a vector. Vectors have a capacity defined whereas there is no such capacity defined for Lists.

7. What is the syntax of declaring a forward_list?

- a) forward_list f;
- b) forward_list<type> f;
- c) forward_list f<type>;
- d) forward_list<type,size> f;

View Answer

Answer: b

Explanation: forward_list<type> f; is the correct syntax of declaring a forward-list.

-
- c) Error
 - d) Segmentation fault

View Answer

Answer: a

Explanation: The program is syntactically correct therefore no error and also memory are handled carefully therefore no segmentation fault. Hence the program runs perfectly. The insert_after() function inserts the elements provided at the position mention in the first argument.

-
- a) 1 2 3 4 5
 - b) 1 3 4 5
 - c) 2 3 4 5
 - d) 1

View Answer

Answer: b

Explanation: erase_after() function is used to erase/delete the element present next to the provided element. So in the given program we provided ffl.begin() i.e. 1 as the element to erase_after() function hence the element after 1 i.e. 2 is deleted.

View Answer

Answer: c

Explanation: remove_if() function is provided in list to remove element based on the conditions provided in the function. So in the program we asked to delete all the element which are greater then 3, hence 4 and 5 are deleted and we are remained with 1,2 and 3.

output:

```
$ ./a.out
1 2 3 4 5
1 2 3
```

-
- a) 1 2 3 4 5
 - b) 1 2 3 4 5 6 7 8 9 10
 - c) 1 7 8 9 10
 - d) 2 3 4 5 6

View Answer

Answer: b

Explanation: splice_after() function is used to insert a forward-list into another list after a given position. So in this program we are trying to insert list2 into list1 after ffl.begin() i.e. 1. Hence the list1 becomes 1 2 3 4 5 6 7 8 9 10.

Output:

```
$ ./a.out  
1 2 3 4 5 6 7 8 9 10
```

ayzom.com

C++ Programming Questions and Answers – STL – Pair

1. What is a pair?

- a) Container consisting of two data elements of the same type
- b) Container consisting of two data elements of different type
- c) Container consisting of one header and two data elements of the same type
- d) Container consisting of two data elements can have the same or different type

View Answer

Answer: d

Explanation: Pair is a container defined in STL which consist of two elements which can be of same or different types.

2. Which header file is required to use pair container in your program?

- a) <algorithm>
- b) <utility>
- c) <pair>
- d) <utilityPair>

View Answer

Answer: b

Explanation: Pair container is defined under the header file <utility> therefore one should include header before using pair container.

3. Which of the following is the correct syntax of using pair p?

- a) pair <type,type> p;
- b) pair p <type,type>;
- c) pair [type,type] p;
- d) pair p [type,type];

View Answer

Answer: a

Explanation: A pair is declared using the this syntax pair <type, type> identifier.

4. Which of the following operations can be performed on a pair?

- a) Assignment of pairs
- b) Copying of one pair to another
- c) Comparison of two pairs
- d) All of the mentioned

View Answer

Answer: d

Explanation: A pair can be assigned, copied or can be compared. Hence all the above operations can be performed on pairs.

5. Which operator is used to access the first or second element of a pair?

- a) ->
- b) .
- c) *
- d) []

View Answer

Answer: b

Explanation: .(dot) operator is used to access the first or second element of a pair. For example, if p = (1,2) is a pair then 1 can be accessed by using p.first and 2 can be accessed using p.second.

6. Which of the following is the correct syntax of accessing the first element of a pair p?

- a) p.first

b) p.second
c) p[0]
d) p[1]
View Answer

Answer: a

Explanation: To access the first element of a pair we use first. for example, if p = (1,2) is a pair then we will use p.first to access the first element of the pair.

7. Which of the following is the correct syntax of accessing the second element of a pair p?

a) p.first
b) p.second
c) p[0]
d) p[1]
View Answer

Answer: b

Explanation: To access the second element of a pair we use second. for example, if p = (1,2) is a pair then we will use p.second to access the second element of the pair.

a) Pair(first,second) = (1,2)
b) Compile-time error
c) Run-time error
d) Assignment is not correct
View Answer

Answer: a

Explanation: This is a way of assigning a pair therefore the program is correct hence the program runs perfectly and outputs the value as follows.
Output:

```
$ ./a.out  
Pair(first,second) = (1,2)
```

a) Pair(first,second) = (1,2)
b) Compile-time error
c) Run-time error
d) Assignment is not correct
View Answer

Answer: b

Explanation: A pair always expects template arguments i.e. types of first and second during declaration of pair. In this program as we have not mentioned the template arguments i.e. types of first and second therefore the program gives an error.

a) Pair(first,second) = (1,2)
b) Compile-time error
c) Run-time error
d) Assignment is not correct
View Answer

Answer: a

Explanation: make_pair() is a function provided to define the values for a pair. Hence the program is correct therefore the program runs successfully.

Output:

```
$ ./a.out  
Pair(first,second) = (1,2)
```

11. Which of the following is correct way of copying the values of pair p1 into other pair p2?

- a) pair <type,type> p2 = p1;
- b) pair <type,type> p2(p1);
- c) Both pair <type,type> p2 = p1; and pair <type,type> p2(p1);
- d) Pair <int,int> p2.copy(p1);

View Answer

Answer: c

Explanation: Both pair <type,type> p2 = p1; and pair <type,type> p2(p1); can be used to copy the data of one pair into other pair.

12. What happens if a pair is not initialized?

- a) Both first and second part is initialized to zero or null
- b) Both first and second part is initialized a garbage value
- c) First is initialized to zero or null and second is initialized a garbage value
- d) Second is initialized to zero or null and first is initialized a garbage value

View Answer

Answer: a

Explanation: If a pair is not initialized then by default both parts of the pair is initialized to zero.

13. Which of the following Operator cannot be used with pairs?

- a) +
- b) ==
- c) =
- d) !=

View Answer

Answer: a

Explanation: We can use only assignment and logical operators with pairs.

View Answer

Answer: a

Explanation: Initially the pair p1 = (1,2) therefore Pair(first,second) = (1,2) is printed and when we have used swap function to swap p1 with p2 the p1 and p2 is swapped therefore next time Pair(first,second) = (3,4) is printed.

Output:

```
$ ./a.out
Pair(first,second) = (1,2)
Pair(first,second) = (3,4)
```

- a) P1 is small
- b) P2 is small
- c) Error
- d) Segmentation fault

View Answer

Answer: a

Explanation: As both the elements are small in p1 pair, therefore, the pair p1 is considered small hence the output is as follows.

Output:

```
$ ./a.out
P1 is small
```

C++ Programming Questions and Answers – STL Container Any – 1

1. What is any in C++?

- a) STL container used to store a single value of any type
- b) Exception class in C++
- c) Fundamental type provided by C++
- d) Template data type

View Answer

Answer: a

Explanation: Any is an STL container provided by C++ to store value or objects of any type.

2. In how many different ways any-container can be constructed?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: c

Explanation: There are three basic ways of constructing any variable. They are done using copy initialization, using the constructor or using an assignment operator.

d) any <type>variable_name = value;

View Answer

Answer: a

Explanation: To initialize an any variable using copy initialization we use the following syntax:

```
any variable_name = value;
```

d) any <type>variable_name = value;

View Answer

Answer: b

Explanation: To initialize an any variable using parameterized constructor we use the following syntax:

```
any variable_name(value);
```

d) any <type>variable_name = value;

View Answer

Answer: b

Explanation: To initialize an any variable using assignment operator we use the following syntax:

```
any variable_name;  
variable_name = value;
```

6. Which of the following syntax is used to convert any variable to its original type?

- a) any_cast<variable_name>();
- b) any_cast(variable_name);
- c) <original_type>(variable_name);
- d) any_cast<original_type>(variable_name);

View Answer

Answer: d

Explanation: The syntax used to convert the any variable to its original type is as follows:

```
any_cast(variable_name);
```

7. Which header file is required to use any container?

- a) <any>
- b) <stl>
- c) <container-any>
- d) <containers>

View Answer

Answer: a

Explanation: <any> header file is required to use any container and its related functions.

- a) 5
- b) Compile-time error
- c) Run-time error
- d) Nothing is printed

View Answer

Answer: b

Explanation: C++ does not allow programmer to directly print the value of any container variable. One need type cast the any variable before printing.

- a) Run-time error
- b) Compile-time error
- c) Hello World
- d) Nothing is printed

View Answer

Answer: d

Explanation: In the above program as we have converted the value to its original type before printing therefore the program runs perfectly and outputs "Hello World".

- a) Hello World
- b) Compile-time error
- c) Run-time error
- d) Nothing is printed

View Answer

Answer: c

Explanation: In this program as we are trying to convert an string into char* which is not same therefore the program gives run-time error saying bad_any_cast.

C++ Programming Questions and Answers – STL Container Any – 2

1. Which exception is thrown if the typecasting is not done properly?

- a) bad_type_cast
- b) bad_any_cast
- c) type_mismatched
- d) any of the mentioned

View Answer

Answer: b

Explanation: bad_any_cast exception is thrown when typecasting is not done properly by the user i.e. if any is storing int value and we are trying to cast it into a string then the program will throw bad_any_cast exception.

2. What is the use of emplace() function?

- a) Used to change the object any container is holding
- b) Used to add more item to the any list
- c) Used to empty any container value
- d) Used to check the type of any variable

View Answer

Answer: a

Explanation: emplace() function is used to change the object contained in any container i.e destroying the present object and creating the new object for the value given by the user.

d) Error

View Answer

Answer: c

Explanation: In this program we are using emplace() function to change the any variable contents and this is allowed in C++ therefore the program runs fine.

4. What is the use of type() function in any container?

- a) Used to destroys the contained object in any variable
- b) Used to change the object any container is holding
- c) Used to return the type information about the any container
- d) Used to check whether a container is empty or not

View Answer

Answer: c

Explanation: type() function is used to check the type of data/value the container object is holding.

- a) f
- b) d
- c) Pkc
- d) u

View Answer

Answer: a

Explanation: The type function is used to get information about the data stored in the any container variable. name() attribute is used to print the type id of the data. Now as the data stored in any variable is float therefore the program outputs f as f is the type id for float.

6. What is the use of has_value() function in any container?

- a) Used to destroys the contained object in any variable
- b) Used to change the object any container is holding
- c) Used to return the type information about the any container

d) Used to check whether any container is empty or not

View Answer

Answer: d

Explanation: has_value() function is provided to check whether a given any container is empty or not.

a) Var is Empty

b) Var is not Empty

c) Error

d) Segmentation fault

View Answer

Answer: b

Explanation: As the variable is containing the information about the float value val = 5.5 therefore the container is not empty therefore the program outputs "Var is not Empty".

8. What is the use of reset() function?

a) Used to destroys the contained object in any variable

b) Used to change the object any container is holding

c) Used to empty any container value

d) Used to check the type of any variable

View Answer

Answer: a

Explanation: reset() function is provided with any to destroy an object contained in any variable in case it is not needed.

c) 5.5

d) 5.5

var is empty

View Answer

Answer: d

Explanation: As the program uses reset() function which resets/destroys an object contained inside the any container therefore var becomes empty hence the program outputs "var is empty".

10. In how many ways we can handle errors in any class?

a) 1

b) 2

c) 3

d) 4

View Answer

Answer: b

Explanation: There are two ways of handling errors in any container first by using exceptions like bad_any_cast and second by returning the pointer.

C++ Programming Questions and Answers – STL – Heap

1. Which type of heap is implemented in STL heap?

- a) max heap
- b) min heap
- c) middle heap
- d) close heap

View Answer

Answer: a

Explanation: C++ STL-heap implements max heap i.e. the front of heap contains the maximum of all the elements in a range.

2. Which function is used to construct heap from given sequence of numbers?

- a) create_heap()
- b) make_heap()
- c) construct_heap()
- d) start_heap()

View Answer

Answer: b

Explanation: C++ STL-heap container provides make_heap() function to convert a given range of number into heap.

3. What is the use of front() function in heap?

- a) Returns the element closest to the median of a sequence
- b) Returns the last element of the heap
- c) Returns the first element of the heap
- d) Returns the element closest to mean of a sequence

View Answer

Answer: c

Explanation: C++ STL-heap container provides the front() function that returns the first element of the heap i.e. the maximum number of the sequence.

4. Which function is used to insert an element into heap?

- a) push_back()
- b) push_heap()
- c) pop_back()
- d) pop_heap()

View Answer

Answer: b

Explanation: C++ STL-heap container provides push_heap() function that inserts a new element to the constructed heap.

5. Elements in STL heap are removed in _____

- a) decreasing order
- b) increasing order
- c) alternate i.e. once max element then min element
- d) input order

View Answer

Answer: a

Explanation: C++ STL-heap simulates the max heap i.e. the maximum element is at the top/front of the heap hence on popping we pop the first element which is always the maximum number in the sequence.

6. Which header file is required to use heap in your program?

- a) <heap>
- b) <algorithm>
- c) <vector>
- d) <map>

View Answer

Answer: b

Explanation: <algorithm> header file is required to use the functionality of the heap container provided by C++.

7. Which of the following is correct syntax of making heap from a vector v?

- a) `make_heap(v.elements);`
- b) `make_heap(v);`
- c) `make_heap(v.end(), v.begin());`
- d) `make_heap(v.begin(), v.end());`

View Answer

Answer: d

Explanation: To construct heap using the vector elements one needs to use the following syntax `make_heap(v.begin(), v.end());` which is taking the iterator to first and last element of the vector using which elements of vector can be accessed and heap can be constructed.

- a) 23
- b) 1
- c) 35
- d) 90

View Answer

Answer: a

Explanation: The heap C++ construct is max heap so when we are trying to print the front of heap the maximum element of the sequence will be printed as that will be at the top of heap.

- a) 90
- b) 1
- c) 110
- d) 23

View Answer

Answer: c

Explanation: In this program we are trying to construct heap using the given vector after which we are inserting 110 into the heap which is now the maximum element in the heap so the answer will be 110.

- a) 90
- b) 1
- c) 110
- d) 23

View Answer

Answer: a

Explanation: In this program we are trying to construct heap using the given vector after which we are inserting 110 into the heap and then we are popping one element from the heap and as 110 is at the top of the heap so it will be popped out and we will have 90 at the top of heap so the answer will be 90.

11. What is the use of `sort_heap()` function in heap?

- a) To sort the elements in the heap into descending order
- b) To sort the elements in the heap into ascending order
- c) To sort the first half of the heap
- d) To sort the second half of the heap

View Answer

Answer: b

Explanation: C++ STL-heap container provides `sort_heap()` function to sort the heap into ascending order which will no longer remain a heap.

12. Which function is used to check whether a given sequence is heap or not?

- a) `sort_heap()`
- b) `is_heap()`
- c) `is_heap_until()`
- d) `check_heap()`

View Answer

Answer: b

Explanation: C++ STL-heap container provides `is_heap()` function to check whether a given sequence of elements represents a heap or not. Descending order of elements represents a valid heap.

13. What is the use of `is_heap_until()` function?

- a) Returns the iterator of the last element of the sequence always
- b) Returns the iterator to the position from where the sequence is a heap
- c) Returns the iterator of the position till that the sequence is a heap
- d) Returns the iterator of the first element of the sequence

View Answer

Answer: c

Explanation: C++ STL-heap container provides `is_heap_until()` function which returns the iterator to the position till the container is a heap. For example, we have 7 5 3 1 10 12 so till 1 the sequence forms a heap so this function will return the iterator to the position of element 1.

- a) 00
- b) 01
- c) 10
- d) 11

View Answer

Answer: b

Explanation: Initially the sequence V is not a heap therefore the function returns 0 after `make_heap()` function the vector is converted into heap therefore the function returns 1 this time.

- a) 90 67 47 35 34 23 4
- b) 90 47 34 23 4 35
- c) 90 47 34 23 4 35 67
- d) 90 47 34 23 4

View Answer

Answer: d

Explanation: `is_heap_till()` returns pointer till the sequence is heap so as vector v is heap till 4 so the iterator of 4 is returned therefore the sequence is printed till 4.

C++ Programming Questions and Answers – vtable and vptr

1. What is vtable in C++?

- a) Lookup table to resolve function calls in dynamic manners
- b) Lookup table to resolve function calls in static manners
- c) Lookup table to see which are the functions available for calls throughout the program
- d) Lookup table to check how many functions are there in the program

View Answer

Answer: a

Explanation: vtable is a lookup table that is used to resolve the function calls in dynamic/late binding manners.

2. Which classes can have vtable?

- a) Classes having friend functions
- b) Classes having virtual functions
- c) Classes having static functions
- d) All classes have a vtable

View Answer

Answer: b

Explanation: Classes having virtual functions only need vtable because in those cases only we need to resolve function calls in a dynamic manner.

3. What is the full form of vtable?

- a) V type table
- b) Vector table
- c) Virtual table
- d) Virtual-vector table

View Answer

Answer: c

Explanation: Full form of vtable is a virtual table. This is called so because it stores the information about virtual functions of a class.

4. What is vptr?

- a) A hidden pointer in a class that points to a virtual table of that class
- b) A hidden pointer in a class that points to virtual functions of that class
- c) A hidden pointer in a class that points to virtual members of the class of that class
- d) A pointer in a class that points to other class

View Answer

Answer: a

Explanation: vptr is a hidden pointer available with classes which are used to point to the virtual table of a class.

5. What is the full form of vptr?

- a) Vector Pointer
- b) Virtual Pointer
- c) V type Pointer
- d) Virtual-vector Pointer

View Answer

Answer: a

Explanation: vptr is abbreviated for a virtual pointer which is used to point virtual tables of a class.

6. vptr is _____

- a) a real pointer
- b) like this pointer of class

- c) passed as a parameter to all functions of class
- d) used to resolve self-references

View Answer

Answer: a

Explanation: Unlike this pointer, vptr is a real pointer that points to the virtual table of a class.

-
- a) 16
 - b) 4
 - c) 8
 - d) 1

View Answer

Answer: d

Explanation: As the class is simple containing no variables therefore no member that requires size so the size of class is 1 hence the output is 1.

-
- a) 16
 - b) 4
 - c) 1
 - d) 8

View Answer

Answer: c

Explanation: In this case the class has two member functions but no variable inside classes which requires space hence the size of class is 1 and so the output.

-
- a) 1
 - b) 4
 - c) 8
 - d) 16

View Answer

Answer: c

Explanation: This class has two virtual functions defined. A class having virtual functions by default has a real pointer vptr. Therefore though not mentioned the class has a real pointer ptr which is of size 8 hence the output is 8. The pointer size differs depending on the system, therefore the output may vary in different systems.

-
- a) Base::function1() and Base::function2()
 - b) Base::function1() and D1::function2()
 - c) D1::function1() and Base::function2()
 - d) D1::function1() and D1::function2() or D2::function1() and D2::function2()

View Answer

Answer: a

Explanation: Base class cannot access the members of derived classes therefore there is no conflict of function in Base class. The vtable of this class will contain two entries for both the virtual functions Base::function1() and Base::function2(). Here Base:: tells that the call will use the definition of Base class.

-
- a) Base::function1() and Base::function2()
 - b) D1::function1() and Base::function2()
 - c) Base::function1() and D1::function2()
 - d) D1::function1() and D1::function2()

View Answer

Answer: b

Explanation: In this program as D1 class is inherited from Base class it inherits both the virtual function of Base class but as the class has not overridden the function2() therefore the Base class definition of function2() will be followed. Therefore the class will have 2 entries corresponding

to each function as D1::function1() because it is defined in Class D1 and Base::function2() because function2() is not defined in Class D1.

- a) Base::function1() and Base::function2()
- b) D2::function1() and Base::function2()
- c) Base::function1() and D2::function2()
- d) D2::function1() and D2::function2()

View Answer

Answer: c

Explanation: In this program as D2 class is inherited from Base class it inherits both the virtual function of Base class but as the class has not overridden the function1() therefore the Base class definition of function1() will be followed. Therefore the class will have 2 entries corresponding to each function as Base::function1() because it is not defined in Class D2 and D2::function2() because function2() is defined in Class D2.

ayzom.com

C++ Programming Questions and Answers – Generators

1. What are the Generators in C++?

- a) An object that generates uniformly distributed numbers
- b) An object that generates a number from a given sequence
- c) An object that generates the smallest number from a given range
- d) An object that generates unique numbers

View Answer

Answer: a

Explanation: Generators are objects that generates uniformly distributed numbers which help in generating random numbers.

2. What are Distributions in C++?

- a) Objects that converts a sequence into a sequence having an ascending order
- b) Objects that converts a sequence into a sequence having specific random variable distribution
- c) Objects that converts a sequence into a sequence having a descending order
- d) Objects that converts a sequence into a sequence having only the smallest, largest and median

View Answer

Answer: b

Explanation: Distributions are objects that convert the sequence generated by the generator into a sequence which has a specific random variable distribution like uniform, normal, binomial, etc.

3. Which header file is used for generating random numbers?

- a) <gen_dist>
- b) <distribution>
- c) <generator>
- d) <random>

View Answer

Answer: d

Explanation: <random> header file is required for using generators and distributions which helps in generating random numbers in a program.

4. What is Pseudo-random number engines?

- a) Uses user input for random number generation
- b) Uses an algorithm that does not require any initial seed to generate random numbers
- c) Uses initial seed based algorithm to generate random numbers
- d) Random number generates depends on the program

View Answer

Answer: c

Explanation: Pseudo-random number engines are used to generate random numbers based on an initial seed provided to them.

5. How many Pseudo-random number engines are there?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: c

Explanation: There are three types of Pseudo-random number engines based on the algorithm they use. They are linear_congruential_engine, mersenne_twister_engine and subtract_with_carry_engine.

6. What is linear_congruential_engine?

- a) Pseudo-random number engine that generates random unsigned integers in the range $[0, 2^{w-1}]$ for some w using Mersenne Twister algorithm
 - b) Pseudo-random number engine that generates random unsigned integers
 - c) Pseudo-random number engine that generates random unsigned integers in the range $[0, 2^{w-1}]$ for some w using lagged Fibonacci generator
 - d) Pseudo-random number engine that generates random signed integers in the range $[0, 2^{w-1}]$ for some w using Mersenne Twister algorithm
- View Answer

Answer: b

Explanation: `linear_congruential_engine` is a simple Pseudo-random number engine that generates random unsigned integer using the basic algorithm.

7. What are different operations are used in Pseudo-random number engines?

- a) `operator()`
- b) `min()`
- c) `max()`
- d) all of the mentioned

View Answer

Answer: d

Explanation: Pseudo-random number engines use three operations `operator()` that generates a random number, `min()` that returns minimum value returned by member `operator()` and `max()` returns the maximum value.

8. What is `mersenne_twister_engine`?

- a) Pseudo-random number engine that generates random unsigned integers
 - b) Pseudo-random number engine that generates random unsigned integers in the range $[0, 2^{w-1}]$ for some w using Mersenne Twister algorithm
 - c) Pseudo-random number engine that generates random unsigned integers in the range $[0, 2^{w-1}]$ for some w using lagged Fibonacci generator
 - d) Pseudo-random number engine that generates random signed integers in the range $[0, 2^{w-1}]$ for some w using Mersenne Twister algorithm
- View Answer

Answer: b

Explanation: `mersenne_twister_engine` is a Pseudo-random number engine that generates random unsigned integers in the range $[0, 2^{w-1}]$ for some w using Mersenne Twister algorithm.

9. What is `subtract_with_carry_engine`?

- a) Pseudo-random number engine that generates random unsigned integers
 - b) Pseudo-random number engine that generates random unsigned integers in the range $[0, 2^{w-1}]$ for some w using Mersenne Twister algorithm
 - c) Pseudo-random number engine that generates random unsigned integers in the range $[0, 2^{w-1}]$ for some w using lagged Fibonacci generator
 - d) Pseudo-random number engine that generates random signed integers in the range $[0, 2^{w-1}]$ for some w using Mersenne Twister algorithm
- View Answer

Answer: c

Explanation: `subtract_with_carry_engine` is a Pseudo-random number engine that generates random unsigned integers in the range $[0, 2^{w-1}]$ for some w using lagged Fibonacci generator.

10. What is a Random number generator?

- a) A generator that generates deterministic random numbers
- b) A generator that generates both non-deterministic random numbers and deterministic random numbers
- c) A generator that generates non-deterministic random numbers
- d) A generator that generates a simple random number

View Answer

Answer: c

Explanation: Random number generator is a random number generator which generates non-deterministic random numbers.

11. What is `random_device`?

- a) A pseudo-random number generator

- b) Time-dependent random number generator
- c) Simple random number generator
- d) A true random number generator

View Answer

Answer: d

Explanation: random_device is a true random number generator not the pseudo random number generator.

12. Which algorithm is used in subtract_with_carry_engine?

- a) Mersenne Twister algorithm
- b) Lagged fibonacci generator algorithm
- c) Either the Mersenne Twister algorithm or Lagged fibonacci generator algorithm
- d) Fibonacci generator

View Answer

Answer: b

Explanation: The algorithm used in subtract_with_carry_engine is lagged fibonacci generator, with state sequences.

13. What is the default random engine?

- a) Random number engine that generates pseudo-random numbers
- b) Mersenne Twister 19937 generator generating 32-bit true random number
- c) Random number engine that generates true random numbers
- d) Mersenne Twister 19937 generator generating 32-bit pseudo random number

View Answer

Answer: a

Explanation: Default random engine is a random number engine that generates pseudo random numbers.

14. What are Engine Adaptors?

- a) Class template that adopts a pseudo-random number generator engine
- b) Class template that adopts a pseudo-random number generator engine to produce numbers with a given numbers of bits
- c) Random number engine that generates pseudo-random numbers
- d) Mersenne Twister 19937 generator generating 32-bit true random number

View Answer

Answer: b

Explanation: Engine adaptor is a class template that adapts a pseudo-random number generator to produce random number having a specific number of bits.

15. How many Engine Adaptors are there in C++?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: c

Explanation: There are three types of Engine adaptors in C++, namely discard_block_engine, independent_bits_engine and shuffle_order_engine.

C++ Programming Questions and Answers – Array Type Manipulation

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    cout<<is_array<int>::value;
    cout<<is_array<char[10]>::value;
    cout<<is_array<string>::value;
    return 0;
}
```

1. Which of the header file is used for array type manipulation?

- a) <array>
- b) <type_traits>
- c) <iostream>
- d) std namespace

View Answer

Answer: d

Explanation: Array type manipulation functions are declared inside the namespace std so you can use namespace std to use these functions.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    cout<<is_array<int>::value;
    cout<<is_array<char[10]>::value;
    cout<<is_array<string>::value;
    return 0;
}
```

2. What is the use of is_array() function in C++?

- a) To check if a variable is array type or not
- b) To check if a variable is 1-D array type or not
- c) To check if a variable is 2-D array type or not
- d) To check if a variable is 1-D or 2-D array type or not

View Answer

Answer: a

Explanation: is_array() function is used to check whether a given variable is of array type or not.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    cout<<is_array<int>::value;
    cout<<is_array<char[10]>::value;
    cout<<is_array<string>::value;
    return 0;
}
```

- a) 010
- b) 101

c) 001

d) 110

[View Answer](#)

Answer: a

Explanation: As int and string are not of array type therefore 0 is printed corresponding to them and char[10] is an array of character of size 10 therefore 1 is printed corresponding to this. Hence answer is 010.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    cout<<is_array<int>::value;
    cout<<is_array<char[10]>::value;
    cout<<is_array<string>::value;
    return 0;
}
```

4. What is the use of is_same() function in C++?

a) To check if a variable is array type or not

b) To check whether two variables have the same characteristics

c) To check if two variable is of array type or not

d) To check whether two variables are different or not

[View Answer](#)

Answer: b

Explanation: is_same() function is used to check whether two variables have the same characteristics or not.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    cout<<is_array<int>::value;
    cout<<is_array<char[10]>::value;
    cout<<is_array<string>::value;
    return 0;
}
```

a) 011

b) 101

c) 010

d) 110

[View Answer](#)

Answer: c

Explanation: In 1st and 3rd case both the variables passed to is_same() function are different whereas for 2nd they are same. Hence the answer is 010.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    cout<<is_array<int>::value;
    cout<<is_array<char[10]>::value;
```

```

        cout<<is_array<string>::value;
        return 0;
}

```

6. What is the use of rank() function in C++?

- a) Returns size of each dimension
- b) Returns how many total elements can be stored in an array
- c) Returns how many elements are in array currently
- d) Returns the dimension of an array

View Answer

Answer: d

Explanation: rank() function returns the rank of the array i.e. the dimension of an array. For example, int arr[10][10] has rank 2.

3. What is the output of the following C++ code?

```

#include <iostream>
#include <string>
using namespace std;
int main()
{
    cout<<is_array<int>::value;
    cout<<is_array<char[10]>::value;
    cout<<is_array<string>::value;
    return 0;
}

```

- a) 111
- b) 123
- c) 321
- d) 121

View Answer

Answer: b

Explanation: In this program first array has the single dimension, second one has two dimensions and third one has three dimension therefore the program prints 123.

3. What is the output of the following C++ code?

```

#include <iostream>
#include <string>
using namespace std;
int main()
{
    cout<<is_array<int>::value;
    cout<<is_array<char[10]>::value;
    cout<<is_array<string>::value;
    return 0;
}

```

8. Which of the following is correct about extent() function?

- a) Returns how many elements are in array currently
- b) Returns the size of the 1st dimension
- c) Returns how many total elements can be stored in an array
- d) Returns the size of a given dimension

View Answer

Answer: d

Explanation: The extent() function takes two parameters one denoting the array other showing the dimension for which the size we want to know.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    cout<<is_array<int>::value;
    cout<<is_array<char[10]>::value;
    cout<<is_array<string>::value;
    return 0;
}
```

a) 101010

b) 102030

c) 302010

d) 102010

View Answer

Answer: b

Explanation: In first cout we are passing 0 and size of first dimension of array is 10 therefore 10 is printed. In following cases we have passed 1 and 2 therefore 20 and 30 are printed respectively.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    cout<<is_array<int>::value;
    cout<<is_array<char[10]>::value;
    cout<<is_array<string>::value;
    return 0;
}
```

10. Which of the following is correct about remove_extent() function?

a) Removes the given dimension from an array

b) Removes the first dimension from the right of the array

c) Removes the first dimension from the left of the array

d) Removes the last dimension from the left of the array

View Answer

Answer: c

Explanation: remove_extent() function removes the first dimension i.e. the first dimension from the given array.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    cout<<is_array<int>::value;
    cout<<is_array<char[10]>::value;
    cout<<is_array<string>::value;
    return 0;
}
```

a) 11

b) 12

c) 21

d) 22

View Answer

Answer: b

Explanation: As we are removing the dimensions from these array and then printing the rank of arrays. Therefore as initially they have 2 and 3 as their rank so after deleting the rank becomes 1 and 2 hence the output is 12.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    cout<<is_array<int>::value;
    cout<<is_array<char[10]>::value;
    cout<<is_array<string>::value;
    return 0;
}
```

12. Which of the following is correct about remove_all_extents() function?

- a) Removes the all dimension from an array
- b) Removes the first dimension from the left of the array
- c) Removes the first dimension from the right of the array
- d) Removes the last dimension from the left of the array

View Answer

Answer: a

Explanation: As the name suggests remove_all_extent() function removes all the dimensions from the array. So rank of array after this operation becomes 0.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    cout<<is_array<int>::value;
    cout<<is_array<char[10]>::value;
    cout<<is_array<string>::value;
    return 0;
}
```

- a) 1
- b) 0
- c) Error
- d) Segmentation fault

View Answer

Answer: b

Explanation: As we have deleted all the dimensions of this array therefore the rank of the array becomes zero hence the output is 0.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    cout<<is_array<int>::value;
    cout<<is_array<char[10]>::value;
    cout<<is_array<string>::value;
    return 0;
}
```

- a) 1010
- b) 1020

c) 2020

d) 2030

[View Answer](#)

Answer: c

Explanation: As we are deleting the first dimension from both the arrays and then printing the extent i.e. size of dimension therefore the answer is 2020 as both the array have 20 as the size of their second dimension.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    cout<<is_array<int>::value;
    cout<<is_array<char[10]>::value;
    cout<<is_array<string>::value;
    return 0;
}
```

a) 20

b) 10

c) Error

d) 0

[View Answer](#)

Answer: d

Explanation: As we have removed all the dimensions from the array therefore the output of extent is 0.

C++ Programming Questions and Answers – Tuples – 1

1. What are the tuples in C++?

- a) Objects that can hold more than one element of different types
- b) Objects that can hold a single element of complex type
- c) Objects that can hold more than one element of the same types
- d) Objects that can hold a single element of fundamental type

View Answer

Answer: a

Explanation: Object that can hold more than one elements having different types. For example, an object holding int, float and char types.

2. Which of the following is correct about tuples?

- a) A tuple can hold more than one element
- b) A tuple can hold elements having different types
- c) Elements of tuples are initialized in order
- d) All of the mentioned

View Answer

Answer: d

Explanation: A tuple can hold more than one element of different types. The order of initialization must be the same as the order of declaration.

3. Which header file is required to use tuples in your program?

- a) <stl>
- b) <array>
- c) <slgorithm>
- d) <tuple>

View Answer

Answer: d

Explanation: <tuple> header file is required to use tuples in your program. This header file contains all the related functions about tuples.

4. Which of the following is the correct way of declaring a tuple?

- a) tuple tp<type1, type2, type3>;
- b) tuple tp = new tuple<type1, type2, type3>;
- c) tuple <type1, type2, type3> tp;
- d) Tuple <type1, type2, type3> tp;

View Answer

Answer: c

Explanation: The correct syntax of declaring tuple is tuple <type1, type2, type3> tp; Lowercase tuple is used to declare to tuples therefore Tuple <type1, type2, type3> tp; is wrong.

5. Which of the following function is used to initialize a tuple?

- a) make()
- b) make_pair()
- c) make_tuple()
- d) make_Tuple()

View Answer

Answer: c

Explanation: make_tuple() function is available under the header file which is used to initialize a tuple.

- a) Nothing is printed
- b) Compile-time error

- c) Run-time error
- d) Exception occurs

View Answer

Answer: b

Explanation: As the order of initialization is different from the order of declaration therefore the program gives compile error.

- a) Nothing is printed
- b) Compile-time error
- c) Run-time error
- d) Exception occurs

View Answer

Answer: a

Explanation: The program is correct hence the program is successfully executed. However nothing is printed because we have written any print statement.

- a) Nothing is printed
- b) Compile-time error
- c) Run-time error
- d) Exception occurs

View Answer

Answer: b

Explanation: In this case the order of initialization is different from the order of declaration therefore the program gives compile error.

9. What is the use of get() function in tuples?

- a) To access an element of a tuple
- b) To print an element of a tuple
- c) To check whether the element of the tuple is empty
- d) To delete an element

View Answer

Answer: a

Explanation: get() function is provided with header file to access an element of a tuple.

View Answer

Answer: d

Explanation: As the tuple contains int, char and string in 0, 1 and 2 position respectively therefore the get<0>, get<1> and get<2> prints 4, 1 and Hello respectively.

C++ Programming Questions and Answers – Tuples – 2

1. Which of the following is correct about `tuple_size`?

- a) Returns the number of elements in a tuple
- b) Returns the maximum sized type element
- c) Returns the total number of bits used by the tuple
- d) Returns the sum of non-string values

View Answer

Answer: a

Explanation: `tuple_size` is used to get the number of elements inside a tuple. For example the `tuple_size` of `tp = {1,4,"hello"}` is 3.

- a) 11
- b) 5
- c) 4
- d) 3

View Answer

Answer: d

Explanation: As the number of elements in the tuple is 3 therefore the `tuple_size` of the tuple is 3 hence the output is 3.

3. Which of the following is correct about `swap()`?

- a) Swaps first element of both tuples
- b) Swaps two tuples
- c) Swaps elements of a tuple alternatively
- d) Swaps last elements of two tuples

View Answer

Answer: b

Explanation: `swap()` function is used to swap two tuples. For example `t1 = {1,2}` and `t2 = {'a','b'}` then after swapping both the tuples becomes `t1 = {'a','b'}` and `t2 = {1,2}`.

View Answer

Answer: a

Explanation: In this program initially the tuples were as given which is printed and after swapping the elements of tuples are swapped therefore the tuples are swapped.

5. What is the use of `tie()` function?

- a) Used to replace elements
- b) Used to delete elements
- c) Used to unpack the values of a tuple
- d) Used to check whether two tuples are the same or not

View Answer

Answer: c

Explanation: `tie()` function of header file is used to unpack the elements of a tuple into different variables.

6. How many variants of `tie()` function is there?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: b

Explanation: There are two variants of `tie()` function one with ignore word and other without ignore word. The ignore word is used to ignore the

unpacking of a particular element.

7. Which word is used to stop the unpacking of a value in a tuple?

- a) stop
 - b) ignore
 - c) cancel
 - d) remain
- View Answer

Answer: b

Explanation: Ignore word is used to ignore the unpacking of some elements of a tuple.

View Answer

Answer: d

Explanation: We have used the tie() function to unpack the values of the tuple. The values are then stored into x, y and z respectively.

- c) Error
 - d) Segmentation fault
- View Answer

Answer: a

Explanation: In this we are ignoring the unpacking of char and therefore the char is not unpacked and we are printing x and y.

10. What is the use of tuple_cat() function?

- a) Takes the union of two tuples
 - b) Takes the intersection of two tuples
 - c) Concatenates two tuples
 - d) Removes elements of the second tuple from first
- View Answer

Answer: c

Explanation: tuple_cat() function is used to concatenate two tuples. The new tuple will contain all the elements of both the tuples.

- a) (0, Hello, 1, World)
 - b) ()
 - c) (0, 1)
 - d) (Hello, World)
- View Answer

Answer: a

Explanation: The tuple_cat() function concatenates the two tuples therefore the two tuples of the program are concatenated and the output contains all the elements of both the tuples.

C++ Programming Questions and Answers – Complex Library – 1

1. Which header file is required to use complex class in your program?

- a) <complex>
- b) <math>
- c) <complex_math>
- d) <algorithm>

View Answer

Answer: a

Explanation: <complex> header file is required to use the functionalities of complex numbers.

2. Which of the following is the correct syntax of declaring a complex number?

- a) complex variable_name<type>;
- b) complex<type> variable_name;
- c) Complex<type> variable_name;
- d) Complex variable_name<type>;

View Answer

Answer: b

Explanation: The correct syntax of declaring a complex number object is complex<type> variable_name.

3. Which function is used to get the real part of the complex number?

- a) img_p()
- b) imag_p()
- c) real()
- d) real_p()

View Answer

Answer: c

Explanation: The real() function is provided by the complex <header> to access the real part of a complex number object.

4. Which function is used to get the imaginary part of the complex number?

- a) real()
- b) imag()
- c) imag_p()
- d) real_p()

View Answer

Answer: b

Explanation: The imag() function is provided by the complex <header> to access the imaginary part of a complex number object.

- a) Complex number is: $3 + 5i$
- b) Complex number is: $5 + 3i$
- c) Complex number is: $9 + 25i$
- d) Complex number is: $3 - 5i$

View Answer

Answer: a

Explanation: The first part in the constructor of complex object denotes real part and second part denotes the imaginary part hence the complex number is $3 + 5i$.

6. Which function is used to get the absolute of a complex number?

- a) ret()
- b) norm()

c) mod()
d) abs()
View Answer

Answer: d

Explanation: abs() function is provided by the header to calculate the absolute value of a complex number.

a) Absolute value is: 4
b) Absolute value is: 5
c) Absolute value is: 3
d) Absolute value is: 5.83095
View Answer

Answer: d

Explanation: In this program we are trying to print the absolute value of a complex number using abs() function of <complex> header.

8. Which function is used to get the argument of a complex number?
a) abs()
b) norm()
c) arg()
d) argu()
View Answer

Answer: c

Explanation: The argument of a complex is calculated using the arg() function of the <complex> header.

a) 1.03038
b) 0
c) Not defined
d) Error
View Answer

Answer: a

Explanation: In this program we are trying to print the argument value of a complex number using arg() function of <complex> header.

10. What is the use of polar function?
a) Used to construct a complex number from the real and imaginary part
b) Used to construct a complex number from magnitude and phase angle
c) Used to construct a complex number from the magnitude and real part
d) Used to construct a complex number from argument and phase angle
View Answer

Answer: b

Explanation: The polar() function of a complex header is used to construct the complex number using the magnitude and phase angle.

C++ Programming Questions and Answers – Complex Library – 2

3. What is the output of the following C++ code?

```
#include <iostream>
#include <complex>
using namespace std;
int main()
{
    complex <double> cn(3.0, 4.0);
    cout<<"Norm is: "<<norm(cn)<<endl;
    return 0;
}
```

- a) Complex number with magnitude 5.83095 and phase angle 1.03038 is: (3,5)
- b) Complex number with magnitude 1.03038 and phase angle 5.83095 is: (3,5)
- c) Complex number with magnitude 5.83095 and phase angle 5.83095 is: (3,5)
- d) Complex number with magnitude 1.03038 and phase angle 1.03038 is: (3,5)

View Answer

Answer: a

Explanation: In this program we are trying to construct a complex number using polar() function of <complex> header.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <complex>
using namespace std;
int main()
{
    complex <double> cn(3.0, 4.0);
    cout<<"Norm is: "<<norm(cn)<<endl;
    return 0;
}
```

2. Which function is used to calculate the norm of a complex number?

- a) abs()
- b) norm()
- c) mod()
- d) square_sum()

View Answer

Answer: b

Explanation: <complex> header provides norm() function to calculate the norm of a complex number.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <complex>
using namespace std;
int main()
{
    complex <double> cn(3.0, 4.0);
    cout<<"Norm is: "<<norm(cn)<<endl;
    return 0;
}
```

- a) 9
- b) 16
- c) 25
- d) 5

View Answer

Answer: c

Explanation: In this program we are trying to calculate the norm of a complex number using norm() function of <complex> header.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <complex>
using namespace std;
int main()
{
    complex <double> cn(3.0, 4.0);
    cout<<"Norm is: "<<norm(cn)<<endl;
    return 0;
}
```

4. Which function is used to calculate the conjugate of a complex number?

- a) conj()
- b) reverse()
- c) opp()
- d) find_conj()

View Answer

Answer: a

Explanation: <complex> header provides conj() function to calculate the conjugate of a complex number.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <complex>
using namespace std;
int main()
{
    complex <double> cn(3.0, 4.0);
    cout<<"Norm is: "<<norm(cn)<<endl;
    return 0;
}
```

- a) Conjugate of $3+(4)i$ is: $3+(4)i$
- b) Conjugate of $3+(4)i$ is: $3-(-4)i$
- c) Conjugate of $3+(4)i$ is: $3-(-4)i$
- d) Conjugate of $3+(4)i$ is: $3+(-4)i$

View Answer

Answer: d

Explanation: The complex conjugate of $a+ib$ is $a-ib$ so conj() of $3+4i$ is $3-4i$ or $\text{conj}(\text{complex}(3,4)) = \text{complex}(3,-4)$.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <complex>
using namespace std;
int main()
{
    complex <double> cn(3.0, 4.0);
    cout<<"Norm is: "<<norm(cn)<<endl;
    return 0;
}
```

6. What is the use of proj() function?

- a) Used to calculate the argument of a complex number
- b) Used to calculate the conjugate of a complex number
- c) Used to calculate the negative of a complex number
- d) Used to calculate the projection of a complex number

View Answer

Answer: d

Explanation: <complex> header provides proj() function to calculate the projection of a complex number $a + ib$.

3. What is the output of the following C++ code?

```

#include <iostream>
#include <complex>
using namespace std;
int main()
{
    complex <double> cn(3.0, 4.0);
    cout<<"Norm is: "<<norm(cn)<<endl;
    return 0;
}

```

- a) proj(3,4) : (3,4)
- b) proj(3,4) : (4,3)
- c) proj(3,4) : (-3,-4)
- d) proj(3,4) : (-3,4)

View Answer

Answer: a

Explanation: In this program we are trying to calculate the projection of a complex number `complex(3,4)` using `proj()` function of <complex> header the answer to which is (3,4).

3. What is the output of the following C++ code?

```

#include <iostream>
#include <complex>
using namespace std;
int main()
{
    complex <double> cn(3.0, 4.0);
    cout<<"Norm is: "<<norm(cn)<<endl;
    return 0;
}

```

8. What is the use of log() function in a complex?

- a) To calculate the log of the imaginary part of a complex number
- b) To calculate the log of the real part of a complex number
- c) To calculate the log of a complex number
- d) To calculate the log of the argument of a complex number

View Answer

Answer: c

Explanation: <complex> header provides log() function to calculate the logarithm of a complex number $a + ib$.

3. What is the output of the following C++ code?

```

#include <iostream>
#include <complex>
using namespace std;
int main()
{
    complex <double> cn(3.0, 4.0);
    cout<<"Norm is: "<<norm(cn)<<endl;
    return 0;
}

```

- a) 1.60944
- b) (1.60944,0.927295)

- c) 0.927295
- d) $1.60944 + 0.927295i$

View Answer

Answer: b

Explanation: In this program we are trying to calculate the logarithm of a complex number `complex(3,4)` using `log()` function of `<complex>` header the answer to which is $(1.60944, 0.927295)$.

3. What is the output of the following C++ code?

```
#include <iostream>
#include <complex>
using namespace std;
int main()
{
    complex <double> cn(3.0, 4.0);
    cout<<"Norm is: "<<norm(cn)<<endl;
    return 0;
}
```

- a) $(3.85374, -27.0168)$
- b) 3.85374
- c) -27.0168
- d) $3.85374 - 27.0168i$

View Answer

Answer: a

Explanation: In this program we are trying to calculate the sine of a complex number `complex(3,4)` using `sin()` function of `<complex>` header the answer to which is $(3.85374, -27.0168)$.

C++ Programming Questions and Answers – Valarray

1. What is Valarray in C++?

- a) Container for efficient operations on an array
- b) Container for efficient printing of the array
- c) Container for efficient value conversion of array
- d) Container for efficient type conversion of array

View Answer

Answer: a

Explanation: Valarray is a special container provided by C++ to hold and perform mathematical operations on array efficiently.

2. Which of the following is correct about Valarray?

- a) Supports element-wise mathematical operations
- b) Slower than a normal array
- c) Harder to use
- d) Can have only integer Valarrays

View Answer

Answer: a

Explanation: Valarray is good at performing mathematical operations. Also, Valarray supports element-wise mathematical operations. They are easier to use and can be of any type.

3. Which header file is required for using Valarray?

- a) <array>
- b) <Valarray>
- c) <stl>
- d) <algorithm>

View Answer

Answer: b

Explanation: <Valarray> header file is required to use the functionalities of Valarrays.

4. What is the use of apply() function in Valarray?

- a) Returns new array after shifting elements by the given number
- b) Returns the summation of all elements of the Valarray
- c) Applies the manipulation provided to all the elements of the array
- d) Returns new array after circular shifting elements by the given number

View Answer

Answer: c

Explanation: <Valarray> header provides apply() function to apply any manipulation passed to the function to all the elements in the Valarray.

d) 15 7 25 6 35

View Answer

Answer: a

Explanation: In this program, we are trying to manipulate the Valarray by adding 5 to all the elements of the Valarray. So first we are printing the original array and then manipulated array.

6. What is the use of sum() function in Valarray?

- a) Applies the manipulation provided to all the elements of the array
- b) Returns the summation of all elements of Valarray
- c) Returns new array after shifting elements by the given number
- d) Returns new array after circular shifting elements by the given number

View Answer

Answer: b

Explanation: <Valarray> header provides sum() function to sum all the elements in the Valarray and returns the total sum.

- a) 20
- b) 53
- c) 12
- d) 63

View Answer

Answer: d

Explanation: In this program we are trying to sum up all the elements of Valarray using the sum() function of the complex header.

8. What is the function of shift()?

- a) Applies the manipulation provided to all the elements of the array
- b) Returns the summation of all elements of Valarray
- c) Returns new array after shifting elements by the given number
- d) Returns new array after circular shifting elements by the given number

View Answer

Answer: c

Explanation: <Valarray> header provides shift() function to shift all the elements of the Valarray by a given number either to the left or to the right.

9. Which of the following is correct about the shift?

- a) Returns new array after shifting elements by the given number
- b) Shifts the elements towards left if the argument supplied is positive
- c) Shifts the elements towards the right if the argument supplied is negative
- d) All of the mentioned

View Answer

Answer: d

Explanation: shift() function is used to shift all the elements of the Valarray by a given number. the elements are shifted towards left if the number is positive and towards the right if the number is negative.

View Answer

Answer: a

Explanation: In this program we are trying to shift elements of Valarray towards left by 2 using the shift() function of the complex header.

View Answer

Answer: b

Explanation: In this program we are trying to shift elements of Valarray towards right by 3 using the shift() function of the complex header.

12. Which of the following is correct about shift() and cshift()?

- 1) shift() makes some values of Valarray equal to 0 after shifting by a non-zero number
 - 2) cshift() makes some values of Valarray equal to 0 after shifting by a non-zero number
- a) 2 only
 - b) 1 only
 - c) Both 1 and 2
 - d) Neither 1 nor 2

View Answer

Answer: b

Explanation: After shifting by some non-zero number the shift() function makes some elements of Valarray equal to zero whereas cshift() does not do such things.

View Answer

Answer: c

Explanation: In this program we are trying to circular shift elements of Valarray towards right by 3 using the cshift() function of the complex header.

14. Which function is used to swap two Valarray?

- a) max()
- b) min()
- c) swap()
- d) change()

View Answer

Answer: c

Explanation: <Valarray> header provides swap() function to swap two Valarrays.

15. Which function is used to print the maximum element from Valarray?

- a) change()
- b) min()
- c) swap()
- d) max()

View Answer

Answer: d

Explanation: <Valarray> header provides max() function to print the maximum element from Valarray.

ayzom.com

C++ Programming Questions and Answers – Bitset – 1

1. What is bitset in C++?

- a) An array of bools consuming one bit per element
- b) Vector of bools
- c) C-like arrays of bool elements
- d) Template class

View Answer

Answer: a

Explanation: Bitset is a collection of bool variables with each element consuming only one bit. They are introduced for efficient use of memories.

2. Which of the following is correct about bitset and vector of bools?

- a) Space consumed by bitset is less than `vector<bool>`
- b) Bitset consume only 1 bit per element
- c) Number of elements in bitset should be known at compile time whereas vector can have a dynamic size
- d) All of the mentioned

View Answer

Answer: d

Explanation: Bitset consumes less space compared to bool vector however the size for bitset is static whereas for bool vector size is dynamic.

3. Which of the following is the limitation of bitset over vector bool?

- a) Space
- b) Size
- c) Type
- d) Speed

View Answer

Answer: b

Explanation: Bitset size is static whereas vector size is dynamic therefore the size of a vector can be increased or decreased which is not possible in bitset.

4. Which operator is used to access the nth bit in a bitset?

- a) `->`
- b) `[]`
- c) `.`
- d) `*`

View Answer

Answer: b

Explanation: `[]` operator is used to access the nth bit of a bitset from the right side. For example, if my bitset b is 1010 then `b[0]` represents 0 and `b[1]` represents 1.

5. How many ways are there for constructing a bitset?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: c

Explanation: There are three ways of constructing a bitset. Direct construction, using integer number and using binary string.

6. Which is the correct syntax of constructing a bitset?

- a) `bitset<size> b;`
- b) `bitset<size> b(12);`
- c) `bitset<size> b(string("1100"));`
- d) all of the mentioned

View Answer

Answer: d

Explanation: All of the above mentioned are correct syntax of constructing a bitset. However each has different way of interpretation.

7. Which of the following is correct way of constructing bitset using integer number?

- a) `bitset<size> b;`
- b) `bitset<size> b(12);`
- c) `bitset<size> b(string("1100"));`
- d) `bitset<size> b(float(12));`

View Answer

Answer: b

Explanation: The correct way of constructing bitset using integer number is as follows:

`bitset<size> b(integer_number);`

8. Which of the following is correct way of constructing bitset using binary string?

- a) `bitset<size> b;`
- b) `bitset<size> b(12);`
- c) `bitset<size> b(string("1100"));`
- d) `bitset<size> b(float(12));`

View Answer

Answer: c

Explanation: The correct way of constructing bitset using binary string is as follows:

`bitset<size> b(string("1100"));`

9. What is the default value of a bitset?

- a) All bits are 0
- b) All bits are 1
- c) Leftmost bit is 0
- d) Rightmost bit is 0

View Answer

Answer: a

Explanation: By default, all the bits of a bitset variable is set to 0 i.e. the value of bitset variable is 0.

10. Which header file is required for using bitset in your program?

- a) `<bit>`
- b) `<bitset>`
- c) `<bits>`
- d) `<BitSet>`

View Answer

Answer: b

Explanation: `<bitset>` header file is required to use the functionality of bitsets in your program i.e. use of binary string in your program.

- a) 00001111
- b) 1111
- c) 11110000
- d) 0000000000001111

View Answer

Answer: a

Explanation: As the size provided to bitset variable is 8, therefore, the number of bits that this variable will have is equal to 8. Now as 15 is 1111 so the least most significant digits will be 1111 to make it represent 15. Hence the answer is 00001111.

- a) 00001111
 - b) Compile-time error
 - c) 11110000
 - d) Run-time error
- View Answer

Answer: d

Explanation: As we are using binary string way of constructing bitset and as 15 is not a binary string therefore the program gives an error.

13. Indexing of bitset variables starts from _____

- a) leftmost bit
- b) rightmost bit
- c) same as in an array
- d) front

View Answer

Answer: b

Explanation: The indexing of bitset variable starts from rightmost bit i.e. if you have b = 1100 as your bitset then b[0] = 0, not 1.

14. What is the use of count() function in bitset?

- a) To count the number of 0's
- b) To count the number of 1's
- c) To count the number of total bits in a bitset
- d) To count the number of low bits

View Answer

Answer: b

Explanation: <bitset> header provides the count() function to count the number of 1's or high bits in the bitset variable.

15. What does size() function returns?

- a) To count the number of 0's
- b) To count the number of 1's
- c) To count the number of total bits used by bitset variable
- d) To count the number of low bits

View Answer

Answer: c

Explanation: <bitset> header provides the size() function to count the number of bits used by the bitset variable. For example, if bitset variable is 1010 then size() function returns 4.

C++ Programming Questions and Answers – Bitset – 2

a) 01

b) 00

c) 10

d) 11

View Answer

Answer: a

Explanation: The test() function is used to check whether a bit is set to one. So as 20 is equivalent to 10100 which has bit at index 1 as 0 and bit at index 2 as 1 therefore the output 0 for test(1) and 1 for test(2).

2. Which of the following is correct about any() function in bitset?

a) Returns true if the number of 1's equal to the number of 0's

b) Returns true if any bit is set 0

c) Returns true if any bit is set 1

d) Returns true only if all bits are 1

View Answer

Answer: c

Explanation: <bitset> header provides the any() function which returns true if any of the bit is set to 1 in the bitset variable.

3. Which of the following is correct about none() function in bitset?

a) Returns true if any bits is set 1

b) Returns true if all bits is set 0

c) Returns true if the number of 1's equal to the number of 0's

d) Returns true only if all bits are 1

View Answer

Answer: b

Explanation: <bitset> header provides the none() function which returns true if none of the bit is set to 1 in the bitset variable.

a) 01

b) 00

c) 10

d) 11

View Answer

Answer: a

Explanation: As 20 is equivalent to 10100 which has two bits which are 1 therefore none() function will return false i.e. 0 and any() function will return true i.e. 1.

5. What is the use of the set() function in bitset?

a) Used to make alternate bits zero

b) Used to make a bit 0 in a bitset

c) Used to make all bits zero

d) Used to set bit(s) in a bitset

View Answer

Answer: d

Explanation: <bitset> header provides the set() function to set bit(s) to 1 in a bitset variable.

6. What happens when no argument is supplied to set() function?

a) All alternate bits are set to 1 in a bitset

b) All bits are set to 0 in a bitset

- c) All bits are set to 1 in a bitset
- d) First bit is set to 1

View Answer

Answer: c

Explanation: When no argument is supplied to set() function i.e. function is called with empty parameters then all the bits of the bitset is set to 1.

7. What happens when only one argument is supplied to set() function?

- a) All bits are set to 1 in a bitset
- b) Bit corresponding to an argument is set to 1
- c) All alternate bits are set to 1 in a bitset
- d) First bit is set to 1

View Answer

Answer: b

Explanation: When only one argument is supplied to set() function then bit corresponding to that index is set to 1.

8. What is the use of reset function in bitset?

- a) Used to make alternate bits zero
- b) Used to make a bit 0 in a bitset
- c) Used to make all bits 1
- d) Used to make a bit(s) 0 in a bitset

View Answer

Answer: d

Explanation: <bitset> header provides the reset() function to set bit(s) to 0 in a bitset variable.

9. What happens when no argument is supplied to reset() function?

- a) All bits are set to 1 in a bitset
- b) All bits are set to 0 in a bitset
- c) All alternate bits are set to 0 in a bitset
- d) First bit is set to 0

View Answer

Answer: b

Explanation: When no argument is supplied to reset() function i.e. function is called with empty parameters then all the bits of the bitset is set to 0.

10. What happens when only one argument is supplied to reset() function?

- a) Bit corresponding to an argument is set to 0
- b) All bits are set to 0 in a bitset
- c) All alternate bits are set to 0 in a bitset
- d) First bit is set to 0

View Answer

Answer: a

Explanation: When only one argument is supplied to reset() function then bit corresponding to that index is set to 0.

C++ Programming Questions and Answers – Bitset – 3

1. What is the use of the flip function in bitset?

- a) Used to flip bit(s) in a bitset
- b) Used to flip a bit in a bitset
- c) Used to flip all bits to 1
- d) Used to flip alternate bits

View Answer

Answer: a

Explanation: <bitset> header provides the flip() function to flip bit(s) in a bitset variable i.e. change the bits in a bitset for example 1100 on flipping becomes 0011.

2. What happens when no argument is supplied to flip() function?

- a) All alternate bits are flipped in a bitset
- b) All bits are flipped to 1 in a bitset
- c) All bits are flipped in a bitset
- d) First bit is flipped

View Answer

Answer: c

Explanation: When no argument is supplied to flip() function i.e. function is called with empty parameters then all the bits of the bitset variable is flipped.

3. What happens when only one argument is supplied to flip() function?

- a) All bits are flipped in a bitset
- b) Bit corresponding to argument bit is flipped
- c) All alternate bits are flipped in a bitset
- d) First bit is flipped

View Answer

Answer: b

Explanation: When only one argument is supplied to flip() function then bit corresponding to that index only is flipped.

- a) 00001101
- b) 11111000
- c) 01111111
- d) 01110001

View Answer

Answer: d

Explanation: ^ operator is used to take xor of two bitset variables i.e. if ith bit in both variables are either 1 or 0 then answer is 0 otherwise answer is 1.

- a) 01110001
- b) 11111000
- c) 01111111
- d) 00001101

View Answer

Answer: b

Explanation: << operator is used to shift bits towards left side so if we have 1111 then on shifting this by 2 i.e. 1111 << 2 will result into 1100.

- a) 00001011
- b) 11111000

- c) 01111111
 - d) 00001101
- View Answer

Answer: a

Explanation: >> operator is used to shift bits towards right side so if we have 1111 then on shifting this by 2 i.e. 1111 >> 2 will result into 0011.

7. Which operator is used as not operator in bitset?

- a) |
- b) &
- c) ~
- d) ^

View Answer

Answer: c

Explanation: ~ operator is used as not operator i.e. the negation of a bit for bitset variables in C++.

8. Which operator is used to take AND of two bitset variables?

- a) ~
- b) &
- c) |
- d) ^

View Answer

Answer: b

Explanation: & operator is used as AND operator for bitset variables in C++. ANDing of two bits are 1 only if both are 1.

9. Which operator is used to take OR of two bitset variables?

- a) ~
- b) &
- c) |
- d) ^

View Answer

Answer: c

Explanation: | operator is used as OR operator for bitset variables in C++. ORing of two bits are 0 only if both are 0.

View Answer

Answer: a

Explanation: In this program we are doing not of b1. AND of b1 and b2 and OR of b1 and b2, answer to them are 10100000, 01111111 and 00001101.

C++ Programming Questions and Answers – Class Relationships

1. What is the class relationship?

- a) A relationship between classes that tells how they are related
- b) A relationship between classes that tells how much power one class has over other class
- c) A relationship between classes that tells which parts of a class is visible to other classes
- d) All of the mentioned

View Answer

Answer: d

Explanation: The Class relationship is a concept that helps us in differentiating how one class is related to other classes, the power of one over other and which part one class can be accessed by other class.

2. How many types of class relationships are there?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: d

Explanation: There are basically four types of class relationships namely Inheritance, Aggregation, Composition and Association relationships between classes.

3. Which type of relationship is modelled by Inheritance?

- a) Is-A relationship
- b) Has-A relationship
- c) Part-Of relationship
- d) Belongs-to relationship

View Answer

Answer: a

Explanation: Inheritance models Is-A type of relationship between classes. This is because in this case derived class inherits all property of the base class and Is-A type of B class.

4. How the relationship is made in Association?

- a) Through the objects of classes
- b) Through constructor
- c) Through destructor
- d) Through class Names

View Answer

Answer: a

Explanation: Association relationship between classes is made using the objects of classes like we have a bank object denoting ABC bank and some objects of Employee class XYZ1, XYZ2 and so on. So, in this case, XYZ1 is an employee in ABC bank so there is a relationship between these two objects.

5. How many types of Association can be there between classes?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: d

Explanation: There can be four types of an association relationship between classes namely one-to-one, one-to-many, many-to-one and many-to-many.

6. Why do we need relationships between classes?

- a) To use the functionality of one class into other
- b) To enhance the communication between classes
- c) To increase code re-usability
- d) All of the mentioned

View Answer

Answer: d

Explanation: Relationships are needed to increase the use of features of one class into the other classes i.e. increasing the re-usability of codes and increasing communication between classes.

7. Composition is also a type of _____ relationship.

- a) Aggregation
- b) Association
- c) Inheritance
- d) Both Aggregation and Association

View Answer

Answer: b

Explanation: Composition is also a type of Both Aggregation and Association relationship. Composition relationships are strong relationships whereas others are a superset of this relationship.

8. Which type of relationship is modelled by Composition?

- a) Is-A relationship
- b) Has-A relationship
- c) Part-Of relationship
- d) Have-A relationship

View Answer

Answer: c

Explanation: Composition models the part-of relationship between classes. In this children cannot exist without a parent, therefore, they are part of each other.

9. Which type of relationship is modelled by Aggregation?

- a) Is-A relationship
- b) Has-A relationship
- c) Part-Of relationship
- d) Have-A relationship

View Answer

Answer: b

Explanation: Aggregation models the has-a relationship between classes. In this children can exist without a parent, therefore, they have a relationship.

10. Which of the following relationships is uni-directional?

- a) Aggregation
- b) Association
- c) Composition
- d) Both Aggregation and Composition

View Answer

Answer: d

Explanation: Both Association and composition are uni-directional relationships. For example, departments can have students but another way around is not possible.

11. In which of the following relationship objects of related classes can occur independently?

- a) Aggregation
- b) Association
- c) Composition
- d) Both Aggregation and Association

View Answer

Answer: d

Explanation: In both Aggregations and Association of objects of related classes can occur independently. For example, an employee can have a bank and a bank can have an employee. Also, Association is a superset of Aggregation, therefore, it also follows the same.

12. In which of the following relationship objects of related classes are strongly dependent?

- a) Aggregation
- b) Association
- c) Composition
- d) Both Composition and Association

View Answer

Answer: d

Explanation: In both Composition and Association, objects of related classes are strongly dependent. For example, books will be destroyed if the library does not exist (talking about objects of classes).

13. Composition is a _____ type of Association relationship.

- a) strong
- b) weak
- c) unnecessary
- d) necessary

View Answer

Answer: a

Explanation: Composition is a strong type of Association relationship because in this case, objects are strongly dependent on each other. For example, human and heart cannot exist without each other.

14. Aggregation is a _____ type of Association relationship.

- a) strong
- b) weak
- c) unnecessary
- d) necessary

View Answer

Answer: b

Explanation: Aggregation is a weak type of Association relationship because in this case objects of related classes can occur independently. For example, bank and employee can occur independently.

15. Which type of relationship is modelled by Association?

- a) Is-A relationship
- b) Has-A relationship
- c) Part-Of relationship
- d) Have-A relationship

View Answer

Answer: b

Explanation: Association models the has-a relationship between classes. Similar to aggregation, in this children can exist without a parent, therefore, they have a relationship.

C++ Programming Questions and Answers – More Containers

1. Which container is used to store elements as key-value pair?

- a) map
- b) multimap
- c) unordered map
- d) all of the mentioned

View Answer

Answer: d

Explanation: C++ provides these three containers(map, multimap and unordered map) to store elements as key-value pair.

2. Which container can have the same keys?

- a) map
- b) multimap
- c) unordered map
- d) set

View Answer

Answer: b

Explanation: C++ provide multimap container that is used to make map that can contain same keys i.e. {a: 5} and {a:10} both can exist.

3. Which container is best to keep the collection of distinct elements?

- a) multimap
- b) heap
- c) set
- d) queue

View Answer

Answer: c

Explanation: C++ provides a set container to store a collection of distinct elements. This container behaves similar to mathematical sets.

4. Which container is used to keep priority based elements?

- a) queue
- b) stack
- c) set
- d) priority queue

View Answer

Answer: d

Explanation: C++ provides priority queue container that stores elements based on their priority. For example, if the absolute value is the priority then -6 will be kept before 4 in the priority queue.

5. Sets are implemented using _____

- a) binary search tree
- b) red black tree
- c) avl tree
- d) heap

View Answer

Answer: a

Explanation: Sets are implemented using the search tree so that we can check the presence of any element to be inserted in $O(\log n)$ time in order to remove conflicts between elements.

6. Unordered map is implemented using _____

- a) binary search tree
- b) red black tree
- c) heap
- d) hash table

View Answer

Answer: d

Explanation: As unordered map has no order of keys therefore hash table is used to store key-value pairs in a hash table.

7. Map is implemented using _____

- a) binary search tree
- b) red black tree
- c) heap
- d) hash table

View Answer

Answer: b

Explanation: The map has some order of stored keys therefore red black tree is used to maintain that order and access the elements as soon as possible.

8. Which of the following is correct about the map and unordered map?

- a) Ordering of keys in maps whereas no such order in the unordered map
- b) Maps are implemented red-black trees whereas unordered map are implemented using hash tables
- c) Average search time in the unordered map is $O(1)$ whereas it is $O(\log n)$ in case of maps
- d) All of the mentioned

View Answer

Answer: d

Explanation: All the above mentioned points are correct about maps and unordered maps. Maps uses red-black tree whereas unordered map uses hash tables therefore the average search time for the unordered map is $O(1)$ whereas it is $O(\log n)$ in case of maps.

9. Which of the following queue container can expand or shrink from both directions?

- a) deque
- b) queue
- c) priority queue
- d) stack

View Answer

Answer: a

Explanation: Deque is a short form for a doubly ended queue which can be expanded and shrunk from any side of the queue either from the front or from the back.

10. Which of the following is correct about map and multimap?

- a) Map can have same keys whereas multimap cannot
- b) Implementation of maps and multimap are different
- c) Multimap can have same keys whereas the map cannot
- d) Average search time of map is greater than multimap

View Answer

Answer: c

Explanation: Multimap is similar to map, the only difference that they have is that in multimap elements can have the same keys where in the map we have only one to one key-value pair correspondence.

C++ Programming Questions and Answers – Standard Library Algorithms

1. What is the header file used for declaring the standard library algorithms?

- a) container
- b) algorithm
- c) library
- d) none of the mentioned

View Answer

Answer: b

Explanation: C++ Standard Library, algorithms are components that perform algorithmic operations on containers and other sequences. For this operation, We have to use header file.

2. Pick out the correct method in the c++ standard library algorithm.

- a) mismatch
- b) maximum
- c) minimum
- d) none of the mentioned

View Answer

Answer: a

Explanation: It is a method in the search operation in standard library algorithms.

3. What is the use of make_heap in the heap operation?

- a) Rearrange a heap
- b) Deform a heap
- c) Form a heap
- d) None of the mentioned

View Answer

Answer: c

Explanation: It is used to rearranges a range so that it becomes a heap.

- a) 6
- b) 7
- c) 8
- d) 9

View Answer

Answer: c

Explanation: In this program, We used the union function to find the number of elements.

Output:

```
$ g++ sla.cpp
$ a.out
8
```

- a) 3 4
- b) 3 4 4
- c) 3 4 & 3 4 4
- d) 3 4 4 0

View Answer

Answer: d

Explanation: In this program, We filled out the vector values by using criteria in the for loop.

Output:


```
$ g++ sla1.cpp
$ a.out
3 4 4 0
```

- a) 1 2 3 4 5
- b) 5 4 3 2 1
- c) 0 1 2 3 4
- d) None of the mentioned

View Answer

Answer: b

Explanation: In this program, We reversed the vector values by using the reverse function.

Output:

```
$ g++ sla2.cpp
$ a.out
5 4 3 2 1
```

- a) 3 3
- b) 3 1
- c) 8
- d) None of the mentioned

View Answer

Answer: a

Explanation: In this program, We are counting the number of 10's and 20's in the myints.

Output:

```
$ g++ sla3.cpp
$ a.out
10 appears 3 times
20 appears 3 times
```

- a) 10, 20, 30, 30, 20, 10, 10, 20
- b) 10, 30, 30, 10, 10
- c) 10, 20, 20, 10, 10, 10, 20
- d) None of the mentioned

View Answer

Answer: b

Explanation: In this program, We are removing all the 20's and then we are printing the remaining.

Output:

```
$ g++ sla4.cpp
$ a.out
10, 30, 30, 10, 10
```

9. What is the type of the first item in the heap?

- a) Bigger than others
- b) Lower than others
- c) Mean value of the heap
- d) None of the mentioned

View Answer

Answer: a

Explanation: In C++, when we say heap we mean max heap and first element of max is bigger than others.

10. Pick out the correct library in the following choices.

- a) Search
- b) Generate
- c) Numeric
- d) All of the mentioned

View Answer

Answer: d

Explanation: These are the available libraries in C++.

ayzom.com

C++ Programming Questions and Answers – Sequences and Containers

1. What kind of iteration does `forward_list` provide in C++?

- a) Uni-directional
- b) Bi-directional
- c) Multi-directional
- d) None of the mentioned

View Answer

Answer: a

Explanation: The `forward_list` uses singly linked list hence it is uni-directional. In the `forward_list`, the container provides insertion and removal at anywhere in the program.

2. What does the size of the vector refers to in c++?

- a) Size of vector
- b) Type of vector
- c) Number of elements
- d) None of the mentioned

View Answer

Answer: c

Explanation: In vectors, by size we mean the number of elements in that vector array.

3. Subsequent elements are moved in terms of _____ when an element is inserted in vector?

- a) Assignment Operator
- b) Copy constructor
- c) Both assignment operator and copy constructor
- d) None of the mentioned

View Answer

Answer: c

Explanation: The vector maintains a certain order of its elements, so that when a new element is inserted at the beginning or in the middle of the vector, Subsequent elements are moved backwards in terms of their assignment operator or copy constructor.

- a) 1 2 3
- b) 0 1 2
- c) 1 2 3 4
- d) None of the mentioned

View Answer

Answer: b

Explanation: In this program, We are pushing the values into the vector from 0 to 3 by using for loop.

Output:

```
$ g++ seqc.cpp
$ a.out
0 1 2
```

- a) 78
- b) 16
- c) 94
- d) None of the mentioned

View Answer

Answer: c

Explanation: In this program, We added all the values in the vector by using front and back operation.

Output:

```
$ g++ seqc1.cpp
$ a.out
94
```

- a) 1 3 6
- b) 8 9
- c) 1 3 6 8 9
- d) None of the mentioned

View Answer

Answer: c

Explanation: In this program, We are comparing the values in both lists and erasing it according to certain condition.

Output:

```
$ g++ seqc2.cpp
$ a.out
1 3 6 8 9
```

- a) 100
- b) 200
- c) 300
- d) None of the mentioned

View Answer

Answer: d

Explanation: Segmentation fault will occur as we are trying to access the freed memory cell i.e. we are trying to access the element which is already deleted.

- a) 10
- b) 20
- c) 30
- d) 60

View Answer

Answer: d

Explanation: In this program, We are adding all the values in the queue.

Output:

```
$ g++ seqc3.cpp
$ a.out
60
```

9. What is the use of adapter in STL in c++?

- a) To provide interface
- b) To manipulate the data
- c) To extract the data
- d) None of the mentioned

View Answer

Answer: a

Explanation: Adapters are data types from STL that adapt a container to provide specific interface.

10. Which is used to iterate over container?

- a) Associated iterator type
- b) Data type of objects
- c) Return type of variables

d) None of the mentioned

View Answer

Answer: a

Explanation: None.

ayzom.com

C++ Programming Questions and Answers – Function Objects

3. Which header is need to be used with function objects?

- a) <function>
- b) <functional>
- c) <funct>
- d) <functionstream>

1. What does the function objects implement?

- a) operator
- b) operator()
- c) operand
- d) none of the mentioned

View Answer

Answer: b

Explanation: Function objects are objects specifically designed to be used with a syntax similar to that of functions.

3. Which header is need to be used with function objects?

- a) <function>
- b) <functional>
- c) <funct>
- d) <functionstream>

2. What are the two advantage of function objects than the function call?

- a) It contains a state
- b) It is a type
- c) It contains a state & It is a type
- d) None of the mentioned

View Answer

Answer: c

Explanation: A function object can contain state. The second is that a function object is a type and therefore can be used as a template parameter.

3. Which header is need to be used with function objects?

- a) <function>
- b) <functional>
- c) <funct>
- d) <functionstream>

View Answer

Answer: b

Explanation: None.

3. Which header is need to be used with function objects?

- a) <function>
- b) <functional>
- c) <funct>
- d) <functionstream>

a) 10 20

b) 20 30

c) 10 20 30

d) All of the mentioned

View Answer

Answer: c

Explanation: In this program, We are dividing the first with the second by using function objects.

Output:

```
$ g++ funo.cpp
$ a.out
10 20 30
```

3. Which header is need to be used with function objects?

- a) <function>
- b) <functional>
- c) <funct>
- d) <functionstream>

- a) -3
- b) 3 4 5
- c) 3 -4 5
- d) -3 4 5

View Answer

Answer: d

Explanation: In this program, we have passed “numbers + 3” in transform function. The results of the transform function are stored in numbers(3rd parameter in func) array whose size is 3.

Output:

```
$ g++ funo1.cpp
$ a.out
-3 4 5
```

3. Which header is need to be used with function objects?

- a) <function>
- b) <functional>
- c) <funct>
- d) <functionstream>

- a) 335
- b) 225
- c) 334
- d) 224

View Answer

Answer: a

Explanation: In this program, We calculated the number of letters in every string by using function objects.

Output:

```
$ g++ funo2.cpp
$ a.out
335
```

3. Which header is need to be used with function objects?

- a) <function>
- b) <functional>
- c) <funct>
- d) <functionstream>

View Answer

Answer: d

Explanation: Running the program will show above behaviour because we have given the value in for loop as 5 instead of 3.

3. Which header is need to be used with function objects?

- a) <function>
- b) <functional>
- c) <funct>
- d) <functionstream>

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: c

Explanation: In this program, We are calculating the number of negative elements present in the program by using function objects.

Output:

```
$ g++ funo3.cpp
$ a.out
3
```

3. Which header is need to be used with function objects?

- a) <function>
- b) <functional>
- c) <funct>
- d) <functionstream>

9. Which are instances of a class with member function operator() when it is defined?

- a) function objects
- b) member
- c) methods
- d) none of the mentioned

View Answer

Answer: a

Explanation: Function objects are instances of a class with member function operator() defined. This member function allows the object to be used with the same syntax as a regular function call.

3. Which header is need to be used with function objects?

- a) <function>
- b) <functional>
- c) <funct>
- d) <functionstream>

10. How many parameters does a operator() in a function object should take?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: b

Explanation: In the case of binary function objects, this operator() member function will take two parameters.

C++ Programming Questions and Answers – Non Modifying Sequence Algorithms

1. How does a sequence of objects are accessed in c++?

- a) Iterators
- b) Pointers
- c) Both Iterators & Pointers
- d) None of the mentioned

View Answer

Answer: c

Explanation: A range is any sequence of objects that can be accessed through iterators or pointers, such as an array or an instance of some of the STL containers.

2. How many parameters are present in mismatch method in non-sequence modifying algorithm?

- a) 1
- b) 2
- c) 3
- d) 3 or 4

View Answer

Answer: d

Explanation: There are two definitions of mismatch with either three or four parameters. They are first1, last1, first2 and optional predicate.

3. What will happen in 'all_of' method if the range is empty?

- a) Return true
- b) Return false
- c) Return nothing
- d) None of the mentioned

View Answer

Answer: a

Explanation: Returns true if pred returns true for all the elements in the range [first,last) or if the range is empty, and false otherwise.

- a) 40
1024
- b) 50
1024
- c) 20
1024
- d) None of the mentioned

View Answer

Answer: b

Explanation: In this program, We are finding the elements which are mismatching in both the variables.

Output:

```
$ g++ non.cpp
$ a.out
50
1024
```

- a) 10
- b) 25
- c) 40
- d) 55

View Answer

Answer: b

Explanation: In this program, We used `find_if` method and returned the first odd value in the vector.

Output:

```
$ g++ non1.cpp
$ a.out
25
```

a) 1 2 3 4

b) 4 3 2 1

c) 3 4 2 1

d) 4 1 2 3

View Answer

Answer: d

Explanation: In this program, We are rotating the vector values by 3, So it is printing this option.

Output:

```
$ g++ non2.cpp
$ a.out
4 1 2 3
```

a) 10 20 30 20 10

b) 10 20 30

c) 30 20 10

d) None of the mentioned

View Answer

Answer: a

Explanation: In this program, We are printing only the unique values by comparing every value.

Output:

advertisement

```
$ g++ non3.cpp
$ a.out
10 20 30 20 10
```

a) 33

b) 44

c) 22

d) None of the mentioned

View Answer

Answer: a

Explanation: In this program, We are counting the number of 10's and 20's in the program.

Output:

```
$ g++ non4.cpp
$ a.out
33
```

9. To what kind of elements does non-modifying sequence algorithm can be applied?

a) Range

b) Vector

c) List

d) All of the mentioned

View Answer

Answer: a

Explanation: Non-modifying sequence algorithm can be applied to list and vector for example the “find” function can be applied to list and vector.

10. Pick out the incorrect method in non-modifying sequence algorithm?

- a) find-if
 - b) none-of
 - c) any-of
 - d) like
- View Answer

Answer: d

Explanation: None.

ayzom.com

C++ Programming Questions and Answers – Modifying Sequence Algorithms

1. What is the use of middle parameter in rotate method?

- a) Marks the beginning of a sequence
- b) Marks the ending of a sequence
- c) Marks the elements in a sequence
- d) None of the mentioned

View Answer

Answer: c

Explanation: Forward iterator pointing to the element within the range and that can be moved to the first position in the range.

2. What kind of object is modifying sequence algorithm?

- a) Function template
- b) Class template
- c) Method
- d) None of the mentioned

View Answer

Answer: a

Explanation: It is a group of functions and implemented under algorithm header file.

3. How the sequence of objects can be accessed?

- a) Iterators
- b) Pointers
- c) Both Iterators & Pointers
- d) None of the mentioned

View Answer

Answer: c

Explanation: A range is any sequence of objects that can be accessed through iterators or pointers, such as an array or an instance of some of the STL containers.

- a) 5 5 5 5 0
- b) 8 8 8 8 0
- c) 5 8 5 8 0
- d) 5 5 5 5 5

View Answer

Answer: a

Explanation: In this program, We filled up all the vector values by using fill method.

Output:

```
$ g++ msa.cpp
$ a.out
5 5 5 5 0
```

- a) 10
- b) 10 40
- c) 10 99 40 99
- d) None of the mentioned

View Answer

Answer: c

Explanation: In this program, We are swapping the certain values in two vectors by using iter_swap.

Output:

```
$ g++ msa1.cpp
$ a.out
10 99 40 99
```

- a) 21
- b) 41
- c) 61
- d) All of the mentioned

View Answer

Answer: d

Explanation: In this program, We allocated the values to the vector and then by using transform function, We increased the values.

Output:

```
$ g++ msa2.cpp
$ a.out
21 41 61
```

- a) 10 20 30
- b) 10 30 30 10 10
- c) 10 20 30 30
- d) None of the mentioned

View Answer

Answer: b

Explanation: In this program, We removed the values in the vector by using the remove method.

Output:

```
$ g++ msa3.cpp
$ a.out
10 30 30 10 10
```

- a) 10
- b) 20
- c) 30
- d) 40

View Answer

Answer: b

Explanation: In this program, We used the find method to find the value before 20.

Output:

```
$ g++ msa4.cpp
$ a.out
20
```

9. How many kind of operation can be applied to transform method in c++?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: b

Explanation: There are two kinds of operations. They are unary and binary operation.

10. What operator is used to remove the duplicates in the range?

- a))
- b) ^
- c) %
- d) ==

[View Answer](#)

Answer: d

Explanation: The function uses operator== to compare the pairs of elements.

ayzom.com

C++ Programming Questions and Answers – Stored Sequences

1. What is meant by sequence point?

- a) Represent the point of execution in the program
- b) Represent the whole program
- c) Represent the beginning of the program
- d) None of the mentioned

View Answer

Answer: a

Explanation: A sequence point defines any point in a computer program's execution at which it is guaranteed that all side effects of previous evaluations have been performed.

2. Pick out the correct statement about sequence point.

- a) Sequence point will compile the program
- b) Sequence point will resolve all the side effects
- c) Sequence point will save the program for execution
- d) None of the mentioned

View Answer

Answer: b

Explanation: Sequence point is a point in time at which the dust has settled and all side effects which have been seen so far are guaranteed to be complete.

3. In sequence point, how will the overloaded operators behave like?

- a) Function
- b) Objects
- c) Instance variable
- d) None of the mentioned

View Answer

Answer: a

Explanation: None.

4. What do input and output objects support?

- a) Terminated sequences
- b) Extracted sequences
- c) Null-terminated sequences
- d) None of the mentioned

View Answer

Answer: c

Explanation: cin and cout support null-terminated sequences as valid containers for sequences of characters.

5. What kind of execution does sequence point allow?

- a) Non-overlap
- b) Overlap
- c) Concurrent
- d) None of the mentioned

View Answer

Answer: a

Explanation: To resolve all the side-effects in the program, the sequence point should not be overlapped.

a) Hello world

- b) Hello
 - c) Error
 - d) Runtime error
- View Answer

Answer: a

Explanation: In this program, We converted the char values into the string.

Output:

```
$ g++ sts.cpp
$ a.out
Hello world
```

- a) 1 3 5
 - b) 1 3 9
 - c) 1 9 3 7
 - d) 1 9 3 7 5
- View Answer

Answer: d

Explanation: In this program, We are finding the odd values in the sequence.

Output:

```
$ g++ sts1.cpp
$ a.out
1 9 3 7 5
```

8. When does the next sequence point start?

- a) At the beginning
 - b) After a terminating semicolon
 - c) It is a beginning statement
 - d) None of the mentioned
- View Answer

Answer: b

Explanation: None.

C++ Programming Questions and Answers – Heaps

1. What is meant by heap?

- a) Used for fast retrieval of elements
- b) Used for organising the elements
- c) Used for fast retrieval & organising the elements
- d) None of the mentioned

View Answer

Answer: c

Explanation: A heap is a way to organize the elements of a range that allows for fast retrieval of the element.

2. Which value is pointed out first in heap?

- a) Lowest value
- b) Highest value
- c) First value
- d) None of the mentioned

View Answer

Answer: b

Explanation: The element with the highest value is always pointed by first.

3. Which operator is used to compare the elements in heap?

- a) >>
- b) comp
- c) <
- d) Both comp & <

View Answer

Answer: d

Explanation: The elements in the heap are compared using operator < (for the first version), or comp (for the second version).

- a) 5 10
- b) 5 10 15 20
- c) 5 10 15 20 99
- d) None of the mentioned

View Answer

Answer: c

Explanation: In this program, We popped out 30 and pushed 99 and then we are sorting that value, So it is printing it.

Output:

```
$ g++ heap.cpp
$ a.out
5 10 15 20 99
```

- a) 1
- b) 33
- c) 3
- d) 44

View Answer

Answer: b

Explanation: In this program, We are pushing a new value into heap and printing it.

Output:

```
$ g++ heap1.cpp
$ a.out
33
```

- a) 10
- b) 20
- c) 4
- d) 8

View Answer

Answer: a

Explanation: In this program, We are printing the maximum value in the heap.

Output:

```
$ g++ heap2.cpp
$ a.out
10
```

- a) 5 6 7
- b) 5 6 7 9 8
- c) 9 8 7 6 5
- d) None of the mentioned

View Answer

Answer: b

Explanation: In this program, We are partitioning the value by using the `partial_sort` method.

Output:

```
$ g++ heap3.cpp
$ a.out
5 6 7 9 8
```

8. How to protect the heap from affecting the memory?

- a) Avoid using pointers for associating two data structures
- b) Embed pointed child objects into the parent object
- c) Allocate objects in chunks
- d) All of the mentioned

View Answer

Answer: d

Explanation: None.

9. In what form does the STL provides heap?

- a) queue
- b) list
- c) vector
- d) `priority_queue`

View Answer

Answer: d

Explanation: STL does provide a heap in the form of a `std::priority_queue`.

10. How many types are there in binary heaps?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: b

Explanation: There are two types of heaps. They are min and max heap.

ayzom.com

C++ Programming Questions and Answers – Min and Max

1. Which keyword is used to declare the min and max functions?

- a) iostream
- b) string
- c) algorithm
- d) none of the mentioned

View Answer

Answer: c

Explanation: Algorithm header file contain the supporting files needed for the execution of these functions.

2. What kind of functions are min and max in c++?

- a) Type specific
- b) Variable specific
- c) Type & Variable specific
- d) None of the mentioned

View Answer

Answer: a

Explanation: The min/max functions are type specific but they will not force everything to be converted to/from floating point. The functions that will force everything to be converted to/from floating point are fmin/fmax.

3. How many parameters are needed for minmax function?

- a) 1
- b) 2
- c) 3
- d) All of the mentioned

View Answer

Answer: d

Explanation: The “minmax” function can take the following:

- 1 parameter: An initializer_list object.
 - 2 parameters: Values to compare.
 - 2 parameters: An initializer_list object. and comparison function
 - 3 parameters: Values to compare. and comparison function
-

- a) 2z
- b) 2a
- c) Error
- d) None of the mentioned

View Answer

Answer: a

Explanation: In this program, We found the max value in the given value by using max function.

Output:

```
$ g++ max.cpp
$ a.out
max(1, 2) == 2
max('a', 'z') == z
```

- a) 2 9
- b) 2 7
- c) 3 9
- d) 3 5

View Answer

Answer: a

Explanation: In this program, We found out the minimum value and maximum value of a range.

Output:

```
$ g++ max1.cpp
$ a.out
2
9
```

a) 3 6

b) 2 5

c) 2 6

d) 2 4

View Answer

Answer: a

Explanation: In this program, We are finding the upper bound and lower bound values by using lower_bound and upper_bound methods.

Output:

```
$ g++ max2.cpp
$ a.out
3 6
```

a) Error

b) Runtime error

c) 1 m

d) None of the mentioned

View Answer

Answer: c

Explanation: In this program, We are finding the minimum value by using min method.

Output:

advertisement

```
$ g++ max3.cpp
$ a.out
1 m
```

a) 3 and 6

b) 2 and 5

c) 3 and 5

d) 2 and 4

View Answer

Answer: a

Explanation: In this program, We are finding out the equal range in the vector.

Output:

```
$ g++ max4.cpp
$ a.out
3 and 6
```

9. Which function is used to return the minimum element in the range?

a) min

b) minimum

- c) min_element
- d) none of the mentioned

View Answer

Answer: c

Explanation: The min_element is used to compare the range of elements and it can find out the minimum element.

10. Which operator is used to compare the values to find min and max?

- a) <
- b) >
- c) <<
- d) >>

View Answer

Answer: a

Explanation: Less than(<) operator is sufficient to compare any two elements in heap and construct respective min or max heap accordingly.

ayzom.com

C++ Programming Questions and Answers – Permutations

1. What is meant by permutation in c++?

- a) To find all the values in the range
- b) To find all the combination of the range
- c) To find all the values & combination in the range
- d) None of the mentioned

View Answer

Answer: b

Explanation: The permutation is used to find all the combination of numbers in the range.

2. How the different permutations are ordered in c++?

- a) Compare lexicographically to each other elements
- b) By finding the highest element in the range
- c) By finding the lowest element in the range
- d) None of the mentioned

View Answer

Answer: a

Explanation: None.

3. Pick out the correct statement about permutation.

- a) If the function can determine the next higher permutation, Returns false
- b) If the function can determine the next higher permutation, Returns true
- c) If the function can't determine the next higher permutation, Returns true
- d) None of the mentioned

View Answer

Answer: b

Explanation: If the function can determine the next higher permutation, it rearranges the elements as such and returns true.

- a) ter
- b) ert
- c) ret
- d) returns all the combination of ter

View Answer

Answer: d

Explanation: In the program, We used string permutation to find out all the combination.

Output:

```
$ g++ perm.cpp
$ a.out
ter
tre
etr
ert
rte
ret
```

- a) 1 2 3
- b) 3 2 1
- c) 2 1 3
- d) 1 3 2

View Answer

Answer: a

Explanation: In this program, We are doing the permutation in the do while loop and then printing last permuted value.

Output:

```
$ g++ perm1.cpp
$ a.out
1 2 3
```

- a) 0 1 2
1 0 2
 - b) 0 2 1
2 0 1
 - c) 2 1 0
1 2 0
 - d) All of the mentioned
- View Answer

Answer: d

Explanation: In this program, We are finding the permutation without using the permutation function.

Output:

```
$ g++ perm2.cpp
$ a.out
0 1 2
1 0 2
0 2 1
2 0 1
2 1 0
1 2 0
```

- a) 3 5
 - b) 5 3
 - c) 5 3
3 5
 - d) None of the mentioned
- View Answer

Answer: c

Explanation: In this program, We find out the permutation of two numbers by using sort.

Output:

advertisement

```
$ g++ perm3.cpp
$ a.out
5 3
3 5
```

- a) 355
 - b) 535
 - c) 553
 - d) All of the mentioned
- View Answer

Answer: d

Explanation: In this program, We are finding the permutation for the given value.

Output:

```
$ g++ perm4.cpp
$ a.out
```


355
535
553

9. What is the header file for vector permutation?

- a) vector_permutation.h
- b) vector_perm
- c) <algorithm>
- d) vector_permutation

View Answer

Answer: c

Explanation: To use permutation on a vector we can use the “next_permutation” function defined in the header.

10. How many parameters are required for next_permutation?

- a) 1
- b) 2
- c) 2 or 3
- d) 3

View Answer

Answer: c

Explanation: There are 2 or 3 needed for next_permutation. They are first, last and optional comp for comparing the elements.

ayzom.com

C++ Programming Questions and Answers – C Style Algorithms

1. Pick out the in correct type of function in header file.

- a) Partitions
 - b) Sort
 - c) Merge
 - d) Join
- View Answer

Answer: d

Explanation: First three type of options are available in header file.

2. What type of algorithm is not available in creating our own STL style algorithms?

- a) copy_if()
 - b) remove_copy_if()
 - c) sort
 - d) none of the mentioned
- View Answer

Answer: a

Explanation: None.

3. What is meant by hash tables in C++?

- a) Array data structure
 - b) Keyed array data structure
 - c) Data structure
 - d) None of the mentioned
- View Answer

Answer: b

Explanation: In hash table, every value will have a key, So that it can be accessed easily.

- a) 10
 - b) 20
 - c) 30
 - d) 5
- View Answer

Answer: b

Explanation: In this program, We are forming a heap with the vector and then we are popping one element and finding the maximum element in the heap.

Output:

```
$ g++ style.cpp
$ a.out
20
```

- a) 5 10 15
 - b) 20 25 30
 - c) 40 50
 - d) All of the mentioned
- View Answer

Answer: d

Explanation: In this kind of style algorithm, We are finding the elements in the both the vector by using set_union function.

Output:

```
$ g++ style1.cpp
$ a.out
5 10 15 20 25 30 40 50
```

- a) 8 9 10
- b) 10 8 9
- c) 9 8 10
- d) None of the mentioned

View Answer

Answer: a

Explanation: In this style algorithm, We have sorted the elements in the vector by using the sort method.

Output:

```
$ g++ style2.cpp
$ a.out
8 9 10
```

- a) 6 7 8 9
- b) 9 8 6 7
- c) 6 7 9 8
- d) None of the mentioned

View Answer

Answer: c

Explanation: In this program, We are partial sorting the vector by using the partial sort method.

Output:

```
$ g++ style3.cpp
$ a.out
6 7 9 8
```

- a) 5 10
- b) 30 40
- c) 50 40
- d) 5 10 30 40

View Answer

Answer: d

Explanation: In this style algorithm, We are finding the symmetric difference between the vectors and printing it.

Output:

```
$ g++ style4.cpp
$ a.out
5 10 30 40
```

9. What is the use of includes function in c++?

- a) Compares two ranges of data
- b) Compares two sorted ranges of data
- c) Includes a new element in the range
- d) None of the mentioned

View Answer

Answer: b

Explanation: Returns true if the first sorted range contains all the elements in the second sorted range.

10. How many parameters are required for sort_heap function?

- a) 1
- b) 2
- c) 2 or 3
- d) 3

[View Answer](#)

Answer: c

Explanation: There are three parameters required for sort_heap. There are first element in heap and last element in heap and an optional compare.

ayzom.com

C++ Programming Questions and Answers – Iterators and Sequences

1. How many categories of iterators are there in c++?

- a) 2
- b) 4
- c) 5
- d) 3

View Answer

Answer: c

Explanation: There are five types of iterators. They are Output, Input, Forward, Random access and Bi-directional.

2. Which of the following can serve as random-access iterator?

- a) Memory pointer
- b) Object pointer
- c) Class pointer
- d) None of the mentioned

View Answer

Answer: b

Explanation: Because of this, It can serve as any category of iterator.

3. What kind of pattern is iterator pattern?

- a) Design pattern
- b) Sequence pattern
- c) Adapter pattern
- d) All of the mentioned

View Answer

Answer: a

Explanation: Iterator pattern is a design pattern in which an iterator is used to traverse a container and access the container's elements.

- a) 12 21 32 31
- b) 12 21 31 32
- c) 12 21 32
- d) 12 21 31

View Answer

Answer: b

Explanation: In this program, We are using const_iterator to sort the data in the set.

Output:

```
$ g++ itr.cpp
$ a.out
12 21 31 32
```

- a) 5 6 7 8 9
- b) 5 6 8 9 10
- c) 6 7 8 9 10
- d) None of the mentioned

View Answer

Answer: b

Explanation: In this program, We are erasing the values in the vector based on the given condition.

Output:

```
$ g++ itr1.cpp
$ a.out
5 6 8 9 10
```

- a) 30
- b) 40
- c) 50
- d) 60

View Answer

Answer: c

Explanation: In this program, We are printing the sixth element in the list.

Output:

```
$ g++ itr2.cpp
$ a.out
50
```

- a) 10 20 1 2
- b) 10 20
- c) 1 2
- d) 1 10

View Answer

Answer: a

Explanation: In this iterator, We are copying the first list into second and printing it.

Output:

```
$ g++ itr3.cpp
$ a.out
10 20 1 2
```

- a) 20
- b) 100
- c) 5
- d) None of the mentioned

View Answer

Answer: c

Explanation: In this program, We are printing the number of elements in the list by using distance method.

Output:

```
$ g++ itr4.cpp
$ a.out
5
```

9. In which type of semantics does c++ implements iterator?

- a) Memory
- b) Size
- c) Pointer
- d) None of the mentioned

View Answer

Answer: c

Explanation: C++ uses pointer arithmetic/semantic to implement iterators.

10. By using which operator does point to next element is represent in iterator?

- a) ++
- b) —
- c) +-
- d) None of the mentioned

View Answer

Answer: a

Explanation: '++' operator is used to represent the next element in the iterator.

ayzom.com

C++ Programming Questions and Answers – Checked Iterators

1. What is the use of checked iterators?

- a) Overwrite the bounds of your container
- b) Not allow you to overwrite the bounds of your container
- c) It will check the list value
- d) None of the mentioned

View Answer

Answer: b

Explanation: Checked iterators ensure that you do not overwrite the bounds of your container.

2. What will happen if the iterator is unchecked?

- a) Arising of compiler warnings
- b) Unchecked behavior on program
- c) Nothing will execute
- d) Arising of compiler warnings & Unchecked behavior on program

View Answer

Answer: d

Explanation: We will get unchecked behavior on calls to an unchecked function and Calls to the standard function will result in compiler warnings.

3. How many adaptors support the checked iterators?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: b

Explanation: There are two adaptors that support checked iterators. They are `checked_array_iterator` class, `Checked_iterator` class.

- a) 10
- b) 100
- c) Exception
- d) Error

View Answer

Answer: a

Explanation: In this program, We are pushing the 10th element and printing it.

Output:

```
$ g++ chei.cpp
$ a.out
10
```

- a) 50
- b) 100
- c) 150
- d) Exception

View Answer

Answer: a

Explanation: In this program, We are finding the element a and printing it.

Output:


```
$ g++ chei1.cpp
$ a.out
50
```

- a) 10
- b) 45
- c) It will print the multiplied value of the input
- d) Exception

View Answer

Answer: c

Explanation: In this program, We got the input by using istream iterator and we are manipulating in.

Output:

```
$ g++ chei2.cpp
$ a.out
3
4
12
```

- a) 10, 20, 30
- b) 10, 20
- c) 10, 20, 30,
- d) None of the mentioned

View Answer

Answer: c

Explanation: In this program, We are producing the values by vector and printing it by using ostream iterator.

Output:

ayzom.com

advertisement

```
$ g++ chei3.cpp
$ a.out
10, 20, 30,
```

- a) 2
- b) 3
- c) 4
- d) 5

View Answer

Answer: b

Explanation: In this program, We are using the referencing operator and it can print the value currently pointing it.

Output:

```
$ g++ chei4.cpp
$ a.out
3
```

9. What does the checked iterator allow you to find?

- a) Warnings
- b) Compile time error
- c) Run time error
- d) None of the mentioned

View Answer

Answer: c

Explanation: None.

10. What kind of errors do checked iterators detect?

- a) Uninitialized iterators
- b) Initialized iterators
- c) Range access
- d) Both Uninitialized iterators and range access

View Answer

Answer: d

Explanation: Checked iterators can easily detect the errors in Uninitialized iterators as well as Range of access.

ayzom.com

C++ Programming Questions and Answers – Allocators

1. Where are allocators used?

- a) Allocation of memory
- b) Deallocation of memory
- c) Used for pointers
- d) Both Allocation & Deallocation of memory

View Answer

Answer: d

Explanation: Allocators handle all the request for allocation and deallocation of memory for the container.

2. Where are allocators implemented?

- a) Template library
- b) Standard library
- c) C++ code library
- d) None of the mentioned

View Answer

Answer: b

Explanation: Allocators are implemented in C++ standard library but it is used for C++ template library.

3. Which operator is used to allocate the memory?

- a) =
- b) +
- c) new
- d) free

View Answer

Answer: c

Explanation: The default allocator uses operator new to allocate memory.

a) 1 5 5 4

b) 5 5 4 1

c) 1 4 5 5

d) 1 4 5 2

View Answer

Answer: c

Explanation: In this program, We are sorting the array by using the allocator.

Output:

```
$ g++ all.cpp
$ a.out
1 4 5 5
```

a) steve

b) jobs

c) jobs steve

d) steve jobs

View Answer

Answer: d

Explanation: In this program, We are storing the string and retrieving the string by using get_temporary_method.

Output:

```
$ g++ all1.cpp
$ a.out
steve jobs
```

- a) Hai
- b) HaiHai
- c) HaiHaiHai
- d) None of the mentioned

View Answer

Answer: c

Explanation: In this program, We storing the string in the string buffer and then we are printing it.

Output:

```
$ g++ all2.cpp
$ a.out
HaiHaiHai
```

7. Which operator is used to deallocate the memory?

- a) destroy
- b) free
- c) empty
- d) none of the mentioned

View Answer

Answer: b

Explanation: None.

8. Which header file is used to manipulate the allocator?

- a) allocator
- b) memory
- c) object
- d) none of the mentioned

View Answer

Answer: b

Explanation: Because all the memory allocation and deallocation libraries are declared in .

9. What is the use of reference member type in allocator?

- a) Point to an element
- b) Quantities of element
- c) Reference to an element
- d) None of the mentioned

View Answer

Answer: c

Explanation: free() function is used to free the memory used by the program.

eg,

```
int *p = (int*) malloc(sizeof(int)); //allocation of memory.
free(p); // freeing the memory occupied by pointer p.
```

10. What is the correct syntax for declaring an allocator?

- a) template < class T > class allocator;
- b) template < class T > class;
- c) template class allocator;
- d) None of the mentioned

View Answer

Answer: a

Explanation: It is a type of syntax for declaring the allocator.

ayzom.com

C++ Programming Questions and Answers – Iterators

3. What is the output of the following C++ code?

```
#include<iostream>
#include<iterator>
#include<vector>
using namespace std;
int main()
{
    vector<int> ar = { 1, 2, 3, 4, 5 };
    vector<int>::iterator ptr = ar.begin();
    advance(ptr, 2);
    cout << *ptr << endl;
    return 0;
}
```

1. What are Iterators?

- a) STL component used to point a memory address of a container
- b) STL component used for vectors
- c) STL component used to call functions efficiently
- d) STL component used to define template classes

View Answer

Answer: a

Explanation: Iterators are STL components used to point a memory address of a container. They are used to iterate over container classes.

3. What is the output of the following C++ code?

```
#include<iostream>
#include<iterator>
#include<vector>
using namespace std;
int main()
{
    vector<int> ar = { 1, 2, 3, 4, 5 };
    vector<int>::iterator ptr = ar.begin();
    advance(ptr, 2);
    cout << *ptr << endl;
    return 0;
}
```

2. Which function is used increment the iterator by a particular value?

- a) next()
- b) advance()
- c) prev()
- d) move()

View Answer

Answer: b

Explanation: advance() function is used to increment an iterator by a given value.

3. What is the output of the following C++ code?

```
#include<iostream>
#include<iterator>
#include<vector>
using namespace std;
int main()
{
    vector<int> ar = { 1, 2, 3, 4, 5 };
    vector<int>::iterator ptr = ar.begin();
    advance(ptr, 2);
}
```

```
    cout << *ptr << endl;
    return 0;
}
```

- a) 2
- b) 3
- c) 4
- d) 5

View Answer

Answer: b

Explanation: Initially the ptr is pointing to first element of vector and now as we are advancing the iterator by 2 which takes the iterator the value 3. Hence the output is 3.

Output:

```
$ ./a.out
3
```

3. What is the output of the following C++ code?

```
#include<iostream>
#include<iterator>
#include<vector>
using namespace std;
int main()
{
    vector<int> ar = { 1, 2, 3, 4, 5 };
    vector<int>::iterator ptr = ar.begin();
    advance(ptr, 2);
    cout << *ptr << endl;
    return 0;
}
```

- a) 3
- b) 4
- c) Error
- d) Segmentation fault

View Answer

Answer: c

Explanation: advance() function is used to increment/decrement the iterator by a given value so it does not returns any thing. So when we are doing ptr = advance(ptr, 2); we are expecting the advance() function to return some value but as it doesn't returns anything therefore compiler throws an error.

3. What is the output of the following C++ code?

```
#include<iostream>
#include<iterator>
#include<vector>
using namespace std;
int main()
{
    vector<int> ar = { 1, 2, 3, 4, 5 };
    vector<int>::iterator ptr = ar.begin();
    advance(ptr, 2);
    cout << *ptr << endl;
    return 0;
}
```

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: d

Explanation: next() function returns an iterator to the position which is ahead of current iterator by a distance of given value. Hence when we are calling the function next(ptr,3); then we are storing the result into ptr which is now an iterator pointing to 4.

Output:

```
$ ./a.out
4
```

3. What is the output of the following C++ code?

```
#include<iostream>
#include<iterator>
#include<vector>
using namespace std;
int main()
{
    vector<int> ar = { 1, 2, 3, 4, 5 };
    vector<int>::iterator ptr = ar.begin();
    advance(ptr, 2);
    cout << *ptr << endl;
    return 0;
}
```

6. How many types of Iterators are there?

- a) 5
- b) 2
- c) 3
- d) 4

View Answer

Answer: a

Explanation: There are 5 types of iterators discussed under STL namely:

- i) Input Iterators
- ii) Output Iterators
- iii) Forward Iterators
- iv) Bi-directional Iterators
- v) Random-access Iterators

3. What is the output of the following C++ code?

```
#include<iostream>
#include<iterator>
#include<vector>
using namespace std;
int main()
{
    vector<int> ar = { 1, 2, 3, 4, 5 };
    vector<int>::iterator ptr = ar.begin();
    advance(ptr, 2);
    cout << *ptr << endl;
    return 0;
}
```

7. Pick the correct statement.

- a) Input iterator moves sequentially forward
- b) Input iterator moves sequentially backward
- c) Input iterator moves in both direction
- d) All of the mentioned

View Answer

Answer: a

Explanation: By definition Input iterators move sequentially forward.

3. What is the output of the following C++ code?

```
#include<iostream>
#include<iterator>
#include<vector>
using namespace std;
int main()
{
    vector<int> ar = { 1, 2, 3, 4, 5 };
    vector<int>::iterator ptr = ar.begin();
    advance(ptr, 2);
    cout << *ptr << endl;
    return 0;
}
```

8. Which of the following is correct about Input Iterators?

- a) Input iterators can be used with all relational operators
- b) Input iterators can work with arithmetic operators
- c) No value can be assigned to the location pointed by Input Iterator
- d) All of the mentioned

View Answer

Answer: c

Explanation: Values cannot be assigned to the location pointed by input operators. Input operators cannot be used with all relational and arithmetic operators.

3. What is the output of the following C++ code?

```
#include<iostream>
#include<iterator>
#include<vector>
using namespace std;
int main()
{
    vector<int> ar = { 1, 2, 3, 4, 5 };
    vector<int>::iterator ptr = ar.begin();
    advance(ptr, 2);
    cout << *ptr << endl;
    return 0;
}
```

9. Which of the following is correct about Input Iterators?

- a) They cannot be decremented
- b) Cannot be used in multi-pass algorithms
- c) Can only be incremented
- d) All of the mentioned

View Answer

Answer: d

Explanation: Input iterators can only be incremented and as it cannot be decremented it can be used in single-pass algorithms only.

3. What is the output of the following C++ code?

```
#include<iostream>
#include<iterator>
#include<vector>
using namespace std;
int main()
{
    vector<int> ar = { 1, 2, 3, 4, 5 };
    vector<int>::iterator ptr = ar.begin();
    advance(ptr, 2);
    cout << *ptr << endl;
    return 0;
}
```

10. Which of the following is correct?

- a) Input Iterators are used for assigning
- b) Output Iterators are used for assigning
- c) Both Input and Output Iterators are used for accessing
- d) Both Input and Output Iterators are used for assigning

View Answer

Answer: b

Explanation: Input Iterators are used for accessing and Output Iterators are used for assigning.

3. What is the output of the following C++ code?

```
#include<iostream>
#include<iterator>
#include<vector>
using namespace std;
int main()
{
    vector<int> ar = { 1, 2, 3, 4, 5 };
    vector<int>::iterator ptr = ar.begin();
    advance(ptr, 2);
    cout << *ptr << endl;
    return 0;
}
```

- a) Input Iterator
- b) Output Iterator
- c) Both Input and Output Iterator
- d) Neither Input nor Output Iterator

View Answer

Answer: b

Explanation: As `itl` iterator is used for assigning values to the vector elements therefore it is an output iterator.

3. What is the output of the following C++ code?

```
#include<iostream>
#include<iterator>
#include<vector>
using namespace std;
int main()
{
    vector<int> ar = { 1, 2, 3, 4, 5 };
    vector<int>::iterator ptr = ar.begin();
    advance(ptr, 2);
    cout << *ptr << endl;
    return 0;
}
```

- a) i and ii
- b) i only
- c) i, ii and iv
- d) iii and iv

View Answer

Answer: b

Explanation: Input iterators are used for accessing containers, therefore `*i1 = 1` is wrong. Both Input and Output iterators cannot be decremented therefore `i2--` is also wrong. Output iterators cannot be used with `==` operator, therefore, that is also wrong. So only `cout<<*i1` is correct.

3. What is the output of the following C++ code?

```
#include<iostream>
```

```
#include<iterator>
#include<vector>
using namespace std;
int main()
{
    vector<int> ar = { 1, 2, 3, 4, 5 };
    vector<int>::iterator ptr = ar.begin();
    advance(ptr, 2);
    cout << *ptr << endl;
    return 0;
}
```

13. Which of the following is an advantage of Forward iterator over input and output iterator?

- a) Can be used as both accessing and assigning iterator
- b) Forward iterator can be incremented or decremented
- c) Can be used with relational operators also
- d) Can be used with arithmetic operators also

View Answer

Answer: a

Explanation: Forward iterator is a combination of both input and output iterator, therefore, can be used as both accessing and assigning iterator. Just like Input and output iterator this can also be not used with all relational and arithmetic operators and can be incremented only.

3. What is the output of the following C++ code?

```
#include<iostream>
#include<iterator>
#include<vector>
using namespace std;
int main()
{
    vector<int> ar = { 1, 2, 3, 4, 5 };
    vector<int>::iterator ptr = ar.begin();
    advance(ptr, 2);
    cout << *ptr << endl;
    return 0;
}
```

14. What are Bi-directional iterators?

- a) Iterator same as Forward Iterator
- b) Forward Iterator that can be used in both directions
- c) Iterator that can only be used to access the sequence from both sides
- d) Iterator that can only be used to assign the sequence from both sides

View Answer

Answer: b

Explanation: Bi-directional iterator is a type of forward iterators that can be used for both directions access and assign.

3. What is the output of the following C++ code?

```
#include<iostream>
#include<iterator>
#include<vector>
using namespace std;
int main()
{
    vector<int> ar = { 1, 2, 3, 4, 5 };
    vector<int>::iterator ptr = ar.begin();
    advance(ptr, 2);
    cout << *ptr << endl;
    return 0;
}
```

15. What are Random-access Iterators?

- a) Iterators that can be used to access elements at an arbitrary offset position

- b) Same as Bi-directional iterator
 - c) Input iterator with the additional property of random access
 - d) Output iterator with the additional property of random access
- View Answer

Answer: a

Explanation: Random access iterators are those iterators that can be used to access elements at an arbitrary offset position relative to the memory that the iterator points.

ayzom.com

C++ Programming Questions and Answers – STL Algorithms

1. Which of the header file is used to implement algorithms provided by C++ STL?

- a) <algorithm>
- b) <header>
- c) <algorithms>
- d) <Algorithm>

View Answer

Answer: a

Explanation: <algorithm> header is provided by the C++ to use STL algorithms.

- a) All numbers are even
- b) All numbers are not even
- c) Error
- d) Segmentation fault

View Answer

Answer: a

Explanation: In this program, we are using `all_of()` function to check whether all the members of the vector are even or not. As all the numbers in a vector are 10, therefore, the program output “All numbers are even”.

Output:

```
$ ./a.out
All numbers are even
```

3. How many types of sequence operations are provided by the C++ algorithm STL?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: b

Explanation: There are two main types of sequence operations are provided by the C++ algorithm STL namely Non-modifying sequence operations and Modifying sequence operations.

4. Which of the following is a Modifying Sequence Operation?

- a) `all_of()`
- b) `any_of()`
- c) `equal()`
- d) `swap()`

View Answer

Answer: d

Explanation: `swap()` is a Modifying sequence operation. One can observe from the name itself as `equal`, `all_of` and `any_of` are by name states that they will be used for comparison whereas `swap` is trying to modify the sequence by changing locations of a sequence.

5. Which of the following is a Non-modifying Sequence Operation?

- a) `swap()`
- b) `transform()`
- c) `remove()`
- d) `search()`

View Answer

Answer: d

Explanation: search() is Non-modifying sequence operation because while searching we never change anything whereas swapping, transforming and removing involves modifying the sequence in some way.

6. What is the use of random_shuffle() function of STL algorithm?

- a) To generate the random sequence in a range
- b) To generate a sequence in a given range and arrange them in random order
- c) To rearrange given sequence randomly
- d) To select any random number from the given sequence.

View Answer

Answer: c

Explanation: random_shuffle() function is used to re-arrange a given sequence of elements in a range.

View Answer

Answer: a

Explanation: In this program we have made a vector of numbers from 1-10 and then applied random_shuffle() function on that sequence of numbers, hence after applying the function the sequence is arranged in a random sequence which is printed. The order of numbers in second sequence may vary when executed several times.

Output:

```
$ ./a.out
1 2 3 4 5 6 7 8 9 10
5 4 8 9 1 6 3 2 7 10
```

View Answer

Answer: a

Explanation: partition() function is used to separate a given list into two parts one part is stored in the original container and other is stored in the newly passed container. So, in this case, we are using IsOdd() function to separate elements from each other which will separate odd and even numbers and as partition function is taking odd function and returning value to bound container, therefore, the odd number sequence will be stored in the bound container and even number sequence will be stored in the original v container.

Output:

```
$ ./a.out
1 9 3 7 5
6 4 8 2 10
```

View Answer

Answer: a

Explanation: sort() function is used to sort a given sequence of numbers hence the program takes the vector and sorts the numbers of the vector.

Output:

```
$ ./a.out
4 2 10 5 1 8
1 2 4 5 8 10
```

10. What is the property of stable sort function provided by the STL algorithm?

- a) sorts the elements of a sequence in ascending order preserving the relative order of equivalent elements
- b) sorts the elements of a sequence in descending order preserving the relative order of equivalent elements
- c) arranges the sequence randomly preserving the relative order of equivalent elements
- d) same as sort function of STL algorithm

View Answer

Answer: a

Explanation: stable sort is used to sort the elements of a sequence also preserving the relative order of the equivalent elements.

11. What is the property of partial sort function provided by the STL algorithm?

- a) sorts the elements before the middle element in ascending order and remaining elements are left without any specific order
- b) sorts the elements before the middle element in descending order and remaining elements are left without any specific order

- c) sorts the elements after the middle element in ascending order and remaining elements are left without any specific order
- d) sorts the elements after the middle element in descending order and remaining elements are left without any specific order

View Answer

Answer: a

Explanation: partial sort of STL algorithm is used to sort the given elements before the middle element in ascending order without any specific order of elements after the middle element.

- a) found.
- b) not found.
- c) Error
- d) Segmentation fault

View Answer

Answer: a

Explanation: In this program, we are using trying to search element 4 in the given vector using binary search and as element 4 is present in the vector, therefore, the function prints "found!".

Output:

```
$ ./a.out
found!
```

- a) found.
- b) not found.
- c) Error
- d) Segmentation fault

View Answer

Answer: b

Explanation: As the searching function `binary_search()` uses binary search to find an element in a sequence which need a sequence to be in ascending order but as the vector in the program is not in ascending order, therefore, the binary search fails to search element 4 in the vector hence output is "not found."

Output:

```
$ ./a.out
not found.
```

14. Which function can be used to find the sum of a vector container?

- a) `findsum()`
- b) `accumulate()`
- c) `calcsun()`
- d) `checksum()`

View Answer

Answer: b

Explanation: STL provides `accumulate()` function to find the sum of a vector.

15. Which header file is required to use `accumulate()` function?

- a) `<algorithm>`
- b) `<numeric>`
- c) `<vector>`
- d) `<iostream>`

View Answer

Answer: b

Explanation: `<numeric>` header file is required to use `accumulate` function in a program.

C++ Programming Questions and Answers – Functors

1. What are functors in C++?

- a) Objects of a class which are treated as functions
- b) Objects that are used to call the function of other classes
- c) Functions that are called using pointer objects
- d) Functions that are called only once in a program

View Answer

Answer: a

Explanation: Functors are objects of a class that are treated like function i.e. they can also be used as function calls.

2. Which of the following operators are overloaded for functors?

- a) []
- b) ()
- c) <<
- d) >>

View Answer

Answer: b

Explanation: () operator is overloaded to use functor property in a C++ program because this is the only operator used for a function call.

3. What is the correct function prototype of () operator overloading?

- a) return_type operator(arguments)();
- b) return_type operator(arguments);
- c) return_type operator()(arguments);
- d) return_type operator(Class_name)(arguments);

View Answer

Answer: c

Explanation: The correct syntax of overloading a () operator in a class is as follows:

```
return_type operator()(arguments) {}
```

4. Which of the following is correct about Functors?

- a) Functors should not be declared outside the main function
- b) Overloaded operator () function is not a member of the class
- c) Functors should be declared global
- d) Functors have a state

View Answer

Answer: d

Explanation: Functors are objects of a class which also have other members and member functions which can be used to save states of that functors hence functors have a state. Functors can be declared anywhere in a program.

- a) 5
- b) Compile-time error
- c) Run-time error
- d) Nothing is printed

View Answer

Answer: a

Explanation: As we have overloaded the () operator inside the class Print, therefore, the print object is a functor, therefore, can be used as a function which is just printing the value passed as a parameter.

- a) 5

- b) 10
 - c) Error
 - d) Segmentation fault
- View Answer

Answer: b

Explanation: As add_5 is a functor whose x value is initialized as 5 therefore it adds 5 to the value passed as parameter and returns the sum.

- a) Add add_10(10);
 - b) Add add_10(10);
 - c) Add add_10(10);
 - d) Add add_10(5);
- View Answer

Answer: c

Explanation: Given the above class, we can declare an object of this class, which will be a functor that adds 10 to every value passed to it, like this Add add_10(10); where 10 in bracket shows that the variable 10 is initialized to class member x which will be used to add 10 to every argument.

View Answer

Answer: d

Explanation: The program is correct and we have initialized two functors one for printing the int values to command line and other for printing string on console. Therefore the program runs fine.

9. Which of the following is a built-in example of functors in C++?

- a) multiplication<T> f(a1, a2);
 - b) add<T> f(a1, a2);
 - c) subtract<T> f(a1, a2);
 - d) plus<T> f(a1, a2);
- View Answer

Answer: d

Explanation: plus<T> f(a1, a2); is one of the correct in-built functor available.

10. Which of the following header file is required to use in-built functors of C++?

- a) <any>
 - b) <functional>
 - c) <functor>
 - d) <function>
- View Answer

Answer: b

Explanation: <functional> header file is required to use the functionality of in-built functors provided by C++.

11. What are unary functors?

- a) Functors that accepts only one parameter
 - b) Functors that accepts two parameters
 - c) Functors that accepts more than one parameters
 - d) Functors that accepts other than a specific type of parameter
- View Answer

Answer: a

Explanation: Unary functors are those which accepts only one argument as a parameter in a functor.

12. What are binary functors?

- a) Functors that accepts only one parameter
- b) Functors that accepts more than one parameters

- c) Functors that accepts two parameters
- d) Functors that accepts other than a specific type of parameter

View Answer

Answer: c

Explanation: Binary functors are those which accepts two arguments as a parameter in a functor.

13. How many ways are there to use functors?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: b

Explanation: There are two ways of using functors one like function calls and other like normal class objects.

14. Which of the following is a logical unary functor?

- a) `logical_or<T> f;`
- b) `logical_and<T> f;`
- c) `logical_not<T> f;`
- d) `negate<T> f;`

View Answer

Answer: c

Explanation: `logical_and` and `logical_or` requires two arguments to act upon whereas `negate` and `logical_not` requires only one argument but `negate` also produces non-logical results like `negate` of a number, therefore, it is not logical hence `logical_not<T> f;` is an only logical functor.

15. Which of the following is an arithmetic unary functor?

- a) `logical_not<T> f;`
- b) `logical_and<T> f;`
- c) `logical_or<T> f;`
- d) `negate<T> f;`

View Answer

Answer: d

Explanation: `logical_and` and `logical_or` requires two arguments to act upon whereas `negate` and `logical_not` requires only one argument but `logical_not` produces only logical results, therefore, will not work on arithmetic values whereas `negate` works with all types of values.

where `f` is a functor and `arg1` and `arg2` are the arguments required by the functors?

- a) `f.call(arg1, arg2);`
- b) `f.operator()(arg1, arg2);`
- c) `f.operator(arg1, arg2);`
- d) `f.operator(arg1, arg2)();`

View Answer

Answer: b

Explanation: `f(arg1, arg2)` means we are calling the overloaded `()` operator method which can be called using object `f` as follows `f.operator()(arg1, arg2);` In general, `object.operator()(argument_lists);`

C++ Programming Questions and Answers – String Characters

1. Which is an instantiation of the basic_string class template?

- a) Character
- b) String class
- c) Memory
- d) None of the mentioned

View Answer

Answer: b

Explanation: The string class is an instantiation of the basic_string class template.

2. Which character is used to terminate the string?

- a) \$
- b) Null
- c) Empty
- d) None of the mentioned

View Answer

Answer: b

Explanation: A string of characters is stored in successive elements of a character array are terminated by the NULL character.

3. How does the strings are stored in the memory?

- a) Contiguous
- b) Non-contiguous
- c) Null
- d) All of the mentioned

View Answer

Answer: a

Explanation: The characters of a string are stored contiguously in the memory.

- a) Steve jobs
- b) He founded apple
- c) Steve
- d) Steve jobsndended apple.....

View Answer

Answer: d

Explanation: In this program, We are adding characters to the string. So the output will as follows after addition of characters.

Output:

```
$ g++ str.cpp
$ a.out
Steve jobsndended apple.....
```

- a) Test
- b) string
- c) Test string
- d) Error

View Answer

Answer: d

Explanation: In the for loop, We are not allowed to give a numeric value in string, So it is arising an error.

- a) Steve Jobs

- b) Apple
 - c) Jobs Apple Steve
 - d) None of the mentioned
- View Answer

Answer: c

Explanation: In this program, We are adding the characters at the end of the current value.

Output:

advertisement

```
$ g++ str1.cpp
$ a.out
Jobs Apple Steve
```

-
- a) Jobs the apple
 - b) the apple
 - c) Steve
 - d) Jobs
- View Answer

Answer: a

Explanation: In this program, We are finding the substring of the string by using the substr function.

Output:

```
$ g++ str2.cpp
$ a.out
Jobs the apple
```

-
- a) 8
 - b) 10
 - c) 12
 - d) 9
- View Answer

Answer: b

Explanation: In this program, We are finding the length of the string.

Output:

```
$ g++ str3.cpp
$ a.out
10
```

-
9. Where are the strings stored?
- a) Stack
 - b) Heap
 - c) Both Stack & Heap
 - d) None of the mentioned
- View Answer

Answer: c

Explanation: Dynamic strings(dynamic length) are stored in heap and static string(with fixed size) are stored in stack.

-
10. What will happen if a string is empty?
- a) It can't be created
 - b) Raises an error
 - c) It can be used
 - d) None of the mentioned

View Answer

Answer: c

Explanation: An empty string is a character array with the NULL character in the zeroth index position.

ayzom.com

C++ Programming Questions and Answers – Basic String

1. Which header file is used to manipulate the string?

- a) iostream
- b) iomanip
- c) string
- d) container

View Answer

Answer: c

Explanation: To use the string class, We have to use #include header file.

2. How many maximum number of parameters does a string constructor can take?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: c

Explanation: string(other_string, position, count). It is a type of constructor for the string.

3. Which constant member functions does not modify the string?

- a) bool empty()
- b) assign
- c) append
- d) none of the mentioned

View Answer

Answer: a

Explanation: Because bool empty is a constant member function, So it can't be modified.

- a) I like to code in c
- b) I like to code
- c) I like to code in c++
- d) None of the mentioned

View Answer

Answer: b

Explanation: In this program, We are resizing the string by adding + and then we are resizing it to 14.

Output:

```
$ g++ basicst.cpp
$ a.out
I like to code
```

- a) 9
- b) 10
- c) 11
- d) Not Fix

View Answer

Answer: d

Explanation: str.capacity() returns the size of the storage space currently allocated for the string, expressed in terms of bytes and capacity of the string may be equal or greater. So the answer is not fix, depends on the compiler.

Output:

```
$ g++ basicst1.cpp
$ a.out
10
```

- a) apple
 - b) 12
 - c) 23
 - d) Steve jobs founded the
- View Answer

Answer: c

Explanation: In this program, We are finding a string by using the find method.

Output:

```
$ g++ basicst2.cpp
$ a.out
23
```

- a) Steve
 - b) jobs
 - c) St*v* j*bs
 - d) St*v*
- View Answer

Answer: c

Explanation: In this program, We are replacing the vowels with a asterisk by using find_first_of method.

Output:

```
$ g++ basicst3.cpp
$ a.out
St*v* j*bs
```

- a) Steve jo
 - b) Steve jobs
 - c) Steve
jobs
 - d) None of the mentioned
- View Answer

Answer: c

Explanation: In this program, We are breaking up the strings into the form of tokens.

Output:

```
$ g++ basicst4.cpp
$ a.out
Steve
jobs
```

9. What is the difference between unsigned int length() and unsigned int size()?

- a) Returns a different value
 - b) They are same
 - c) Returns a different value but they are same
 - d) None of the mentioned
- View Answer

Answer: b

Explanation: Both of them will return the length of strings in same notations.

10. How many parameters can a resize method take?

- a) 1
- b) 2
- c) 1 or 2
- d) 2

View Answer

Answer: c

Explanation: There can be one or two parameters in resize method. They are string length and an optional new character to be inserted.

ayzom.com

C++ Programming Questions and Answers – C Standard Library

1. Where are standard C libraries defined in C++?

- a) Container
- b) std namespace
- c) list
- d) None of the mentioned

View Answer

Answer: b

Explanation: Every element of the c library is defined within the std namespace.

2. Which of the following have their changes in their declaration related to constness of parameter?

- a) strchr
- b) string
- c) memory
- d) none of the mentioned

View Answer

Answer: a

Explanation: These are the items which will have their change in declaration related to constness of parameter. They are strchr, strpbrk, strchr, strstr, memchr.

3. How many elements does a floating point number is composed of?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: d

Explanation: The floating point number composed of four elements. They are sign, Base, Significand and Exponent.

- a) 12
- b) 12.37
- c) 13
- d) None of the mentioned

View Answer

Answer: b

Explanation: In this program, We are calculating the double value by using the floating point number and we are using the function strtod.

Output:

```
$ g++ cinc.cpp
$ a.out
12.37
```

- a) 10
- b) 20
- c) 40
- d) 30

View Answer

Answer: c

Explanation: In this program, We are searching for the element and then we are printing it.

Output:

```
$ g++ cinc1.cpp
$ a.out
40
```

- a) 23-11
- b) 1123
- c) 2311
- d) None of the mentioned

View Answer

Answer: c

Explanation: In this program, We are finding the absolute value of the n.

Output:

```
$ g++ cinc2.cpp
$ a.out
2311
```

- a) 3.1416
- b) -3.1416
- c) -3.141600
- d) 3.141600

View Answer

Answer: d

Explanation: In this program, We are finding the absolute value of a floating point value.

Output:

```
$ g++ cinc3.cpp
$ a.out
3.141600
```

- a) 7
- b) 3
- c) 4
- d) None of the mentioned

View Answer

Answer: b

Explanation: In this program, We are finding the remainder of a number by using div function.

Output:

```
$ g++ cinc4.cpp
$ a.out
3
```

9. How does the limits.h header file can be represented in C++?

- a) limits
- b) limit
- c) climits
- d) none of the mentioned

View Answer

Answer: c

Explanation: Any standard library of C can be written in C++ by using 'c' in front of header file name and omitting the '.h'. for example <limits.h> can be written as <climits>.

10. Pick out the correct syntax of the header file that can be used with C++.

- a) `#include <float>`
- b) `#include <float.h>`
- c) Both `#include <float>` & `#include <float.h>`
- d) None of the mentioned

View Answer

Answer: b

Explanation: The C header file that is ending with h can only be used in C++.

ayzom.com

C++ Programming Questions and Answers – Output Stream

1. How many groups of output of operation are there in c++?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: b

Explanation: There are two groups of output operation in c++. They are formatted output and unformatted output.

2. Pick out the correct objects about the instantiation of output stream.

- a) cout
- b) cerr
- c) clog
- d) all of the mentioned

View Answer

Answer: d

Explanation: cout, cerr and clog are the standard objects for the instantiation of output stream class.

3. What is meant by ofstream in c++?

- a) Writes to a file
- b) Reads from a file
- c) Writes to a file & Reads from a file
- d) None of the mentioned

View Answer

Answer: a

Explanation: ofstream is a stream class to write on files.

a) Steve jobs

b) A

c) 65

d) 65

View Answer

Answer: d

Explanation: In this program, We are printing the five spaces and then we are printing the value of 65.

Output:

```
$ g++ ous
.cpp
$ a.out
65
```

a) 2c

b) 2b

c) 20

d) 50

View Answer

Answer: b

Explanation: In this program, We are printing the hexadecimal value of the given decimal number by using hex function.

Output:

```
$ g++ ous1.cpp
$ a.out
2b
```

- a) This is an apple
- b) apple
- c) sample
- d) This is a sample

View Answer

Answer: d

Explanation: In this program, We are changing the ap to sam by using the pos function.

Output:

```
$ g++ ous2.cpp
$ a.out
This is a sample
```

- a) 77
- b) -77
- c) – 77
- d) None of the mentioned

View Answer

Answer: c

Explanation: In this program, We are using the internal function and moving the 77 to one position.

Output:

```
$ g++ ous3.cpp
$ a.out
- 77
```

- a) 3.14
- b) 3.14159
- c) Error
- d) None of the mentioned

View Answer

Answer: b

Explanation: In this program, We are using the locale and then assigning the type to the value.

Output:

```
$ g++ ous4.cpp
$ a.out
3.14159
```

9. How many types of output stream classes are there in c++?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: c

Explanation: There are three output stream classes in c++. They are ostream, ofstream and ostrstream.

10. What must be specified when we construct an object of class ostream?

- a) stream
- b) streambuf
- c) memory
- d) none of the mentioned

View Answer

Answer: b

Explanation: If you construct an object of class ostream, you must specify a streambuf object to the constructor.

ayzom.com

C++ Programming Questions and Answers – Input Stream

1. Which operator is used for input stream?

- a) >
- b) >>
- c) <
- d) <<

View Answer

Answer: b

Explanation: The operator of extraction is >> and it is used on the standard input stream.

2. Where does a cin stops it extraction of data?

- a) By seeing a blank space
- b) By seeing (
- c) By seeing a blank space & (
- d) None of the mentioned

View Answer

Answer: a

Explanation: cin will stop its extraction when it encounters a blank space.

3. Which is used to get the input during runtime?

- a) cout
- b) cin
- c) coi
- d) none of the mentioned

View Answer

Answer: b

Explanation: cin is mainly used to get the input during the runtime.

- a) 73
- b) your value + 4
- c) Error
- d) None of the mentioned

View Answer

Answer: c

Explanation: We are not allowed to do addition operation on cin.

- a) 50
- b) Depends on value you enter
- c) Error
- d) None of the mentioned

View Answer

Answer: b

Explanation: In this program, We are getting the input on runtime and manipulating the value.

Output:

```
$ g++ inp.cpp
$ a.out
Enter price: 3
Enter quantity: 4
Total price: 12
```

- a) First input
- b) Clearing cin
- c) Error
- d) None of the mentioned

View Answer

Answer: d

Explanation: In this program, We are getting the input and clearing all the values.

Output:

advertisement

```
$ g++ inpl.cpp
$ a.out
First input: 4
Clearing cin.
All done.
```

-
- a) 100
 - b) t
 - c) It will print what we enter till character t is encountered in the input data
 - d) None of the mentioned

View Answer

Answer: c

Explanation: The program will store all strings entered and will print them only when the character 't' is encountered.

Input >> coding
Input >> is fun
Input >> t

Output:

coding
is fun

8. How many parameters are there in getline function?

- a) 1
- b) 2
- c) 2 or 3
- d) 3

View Answer

Answer: c

Explanation: There are two or three parameters in getline() function. They are a pointer to an array of characters and maximum number of characters and an optional delimiter.

9. What can be used to input a string with blank space?

- a) inline
- b) getline
- c) putline
- d) none of the mentioned

View Answer

Answer: b

Explanation: If a user wants to input a sentence with blank spaces, then he may use the function getline.

10. When will the cin can start processing of input?

- a) After pressing return key

- b) BY pressing blank space
- c) After pressing return key & BY pressing blank space
- d) None of the mentioned

View Answer

Answer: a

Explanation: When you give some input to console the processing of the input starts when the user presses enter/return key.

ayzom.com

C++ Programming Questions and Answers – Formatting

1. Which is used for formatting purpose in c++?

- a) Whitespace
- b) Container
- c) &
- d) Vector

View Answer

Answer: a

Explanation: Whitespace is a term that refers to characters like spaces, tabs, and newlines that are used for formatting purposes.

2. How many number of spaces should be set in default tab?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: d

Explanation: The default number of spaces is 4 in programming.

3. What can be improved by formatting the source code?

- a) Memory
- b) Address
- c) User interface
- d) None of the mentioned

View Answer

Answer: c

Explanation: By formatting your code, the user can easily understand it and can upgrade it without any complexity.

4. Choose the correct formatted code.

- a) `cout << "Hai"`
`<< "Good morning";`
- b) `cout << "Hai" <<`
`"Good morning" << endl;`
- c) `cout<<"Hai";`
- d) none of the mentioned

View Answer

Answer: b

Explanation: If a long line that is broken into pieces is broken with an operator, the operator should be placed at the end of the line, not the start of the next line.

5. Choose the correct formatted code.

- a) `int a = 5;`
- b) `int a=5;`
- c) `int a =5;`
- d) none of the mentioned

View Answer

Answer: a

Explanation: Variable initialization should be done with one space on left and right of equal operator.

6. Choose the correct formatted code.

- a)

```
int main(){  
cout << "5";}
```
- b)

```
int main()  
{  
cout << "5";  
}
```
- c)

```
int main()  
{  
cout << "5";}
```
- d) none of the mentioned

View Answer

Answer: b

Explanation: The braces that tell where a function begins and ends should be aligned with the function name and be on their own lines.

7. Which function allows you to set minimum width for the next input?

- a) setfill
- b) setw
- c) setwidth
- d) none of the mentioned

View Answer

Answer: b

Explanation: setw function of iomanip header file allows you to set minimum width for the next input.

- a) 1,1,2013
- b) 2013
- c) 01/01/2013
- d) None of the mentioned

View Answer

Answer: c

Explanation: In this program, We are using the setw function to print the date in correct format.

Output:

```
$ g++ form.cpp  
$ a.out  
01/01/2013
```

- a) 64100
- b) 48
- c) 345
- d) None of the mentioned

View Answer

Answer: a

Explanation: In this program, We are finding the octal number of given number by using oct function.

Output:

```
$ g++ form1.cpp  
$ a.out  
64100
```

10. What is the use of the function "showbase"?

- a) Indicate the base used
- b) Indicate the variable
- c) Indicate the base used & variable

d) None of the mentioned

View Answer

Answer: a

Explanation: None.

ayzom.com

C++ Programming Questions and Answers – File Streams and String Streams

1. Which operator is used to insert the data into file?

- a) >>
- b) <<
- c) <
- d) None of the mentioned

View Answer

Answer: b

Explanation: You can write information to a file from your program using the stream insertion operator <<.

2. Which function is used to position back from the end of file object?

- a) seekg
- b) seekp
- c) both seekg & seekp
- d) none of the mentioned

View Answer

Answer: a

Explanation: The member function seekg is used to position back from the end of file object.

3. How many objects are used for input and output to a string?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: c

Explanation: The stringstream, ostringstream, and istream objects are used for input and output to a string.

- a) This is sample
- b) sample
- c) Error
- d) Runtime error

View Answer

Answer: d

Explanation: In this program, if the file exist, it will read the file. Otherwise it will throw an exception. A runtime error will occur because the value of the length variable will be “-1” if file doesn’t exist and in line 13 we are trying to allocate an array of size “-1”.

- a) first
- b) second
- c) Returns first 2 letter or number from the entered word
- d) None of the mentioned

View Answer

Answer: c

Explanation: In this program, We are using the sync function to return the first two letters of the entered word.

Output:

```
$ g++ stream.cpp
$ a.out
Enter a word: steve
s
t
```

- a) Done
- b) Error
- c) Runtime error
- d) None of the mentioned

View Answer

Answer: a

Explanation: In this program, We are using the flush function to update the contents in a file.

Output:

advertisement

```
$ g++ stream1.cpp
$ a.out
Done
```

- a) 100
- b) 3.14
- c) 314
- d) All of the mentioned

View Answer

Answer: d

Explanation: In this program, We are printing the given value and manipulating the given value by using endl.

Output:

```
$ g++ stream2.cpp
$ a.out
100
3.14
314
```

- a) dot operator
- b) insertion operator
- c) \$ symbol
- d) none of the mentioned

View Answer

Answer: a

Explanation: This program will stop getting the input, When it occurs the dot(.) operator.

Output:

```
$ g++ stream3.cpp
$ a.out
Steve.
```

9. Which member function is used to determine whether the stream object is currently associated with a file?

- a) is_open
- b) buf
- c) string
- d) none of the mentioned

View Answer

Answer: a

Explanation: The member function is_open can be used to determine whether the stream object is currently associated with a file.

10. Which header file is used for reading and writing to a file?

- a) #include<iostream>

- b) `#include<fstream>`
 - c) `#include<file>`
 - d) None of the mentioned
- View Answer

Answer: b

Explanation: `<fstream>` header file contains all the file reading and writing functions. Also `<ifstream>` and `<ofstream>` contains functions only reading and only writing to files related functions respectively.

ayzom.com

C++ Programming Questions and Answers – Buffering

1. Which one is always faster in writing on C++?

- a) Writing to a file
- b) Writing to memory
- c) Reading from the network
- d) None of the mentioned

View Answer

Answer: b

Explanation: For the stand of file operations, writing to memory (RAM) is always faster than writing to the file on the disk directly.

2. How many tests are available in read and write operations?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: b

Explanation: There are two types of read and write tests. They are throughput in random reads and throughput in contiguous reads.

3. What will act as a intermediate between i/o operations and physical file?

- a) Memory
- b) Ram
- c) Stream buffer
- d) None of the mentioned

View Answer

Answer: c

Explanation: A stream buffer is a block of data that acts as intermediary between the i/o operations and the physical file associated to the stream.

- a) Buffered stream
- b) Unbuffered stream
- c) Error
- d) Buffered & Unbuffered stream

View Answer

Answer: d

Explanation: In this program, the fopen will create the file and send the text to both of the files.

Output:

```
$ g++ buf.cpp
$ a.out
"Buffered stream" in myfile.txt
"Unbuffered stream" in myfile2.txt
```

- a) 10
- b) 20
- c) 5
- d) Depends upon the file

View Answer

Answer: d

Explanation: This program will return the size of the file in bytes.

Output:

```
$ g++ buf1.cpp
$ a.out
20
```

- a) This sentence
- b) This sentence is redirected
- c) This sentence is redirected to a file
- d) None of the mentioned

View Answer

Answer: c

Explanation: In this program, We are sending the text to the file by opening the file using the function freopen.

Output:

```
$ g++ buf2.cpp
$ a.out
```

This sentence is redirected to a file

- a) ABCD
- b) ABC
- c) ABCDE
- d) None of the mentioned

View Answer

Answer: b

Explanation: In this program, We are setting the buffer size upto 3 only, So it is printing ABC.

Output:

```
$ g++ buf3.cpp
$ a.out
ABC
```

- a) xyz
- b) zyx
- c) yxz
- d) none of the mentioned

View Answer

Answer: a

Explanation: In this program, We are writing into the file by using the function fwrite.

Output:

```
$ g++ buf4.cpp
$ a.out
xyz
```

9. By using which function does the buffer are automatically flushed?

- a) fopen
- b) copy
- c) compare
- d) fclose

View Answer

Answer: d

Explanation: fclose() is used to close the file and flush it out of memory for safe closing of files opened during the execution of program. In short to avoid memory faults during the execution of the program.

10. How many parameters are available in the function setbuf?

- a) 1
- b) 2
- c) 3
- d) 4

[View Answer](#)

Answer: b

Explanation: There are two parameters that are used in setbuf. They are stream and buffer.

ayzom.com

C++ Programming Questions and Answers – Locale

1. What is the main feature of locale in C++?

- a) Sustainability
- b) Portability
- c) Reliability
- d) None of the mentioned

View Answer

Answer: b

Explanation: A locale is a set of features that are culture-specific, which can be used by programs to be more portable internationally.

2. Which objects information is loaded in locale object?

- a) facet object
- b) instead object
- c) Both facet & instead object
- d) None of the mentioned

View Answer

Answer: a

Explanation: A locale object contains information about which facet objects constitute a locale, and is each one of these facet objects that implements specific features as member functions.

3. How many categories are available in facets?

- a) 4
- b) 5
- c) 6
- d) 3

View Answer

Answer: c

Explanation: There are 6 categories of facet in c++. They are collate, ctype, monetary, numeric, time and messages.

- a) Steve jobs
- b) STEVE JOBS
- c) Steve
- d) None of the mentioned

View Answer

Answer: b

Explanation: In this program, We have converted the lower case letters to uppercase letters by using toupper function.

Output:

```
$ g++ loc.cpp
$ a.out
STEVE JOBS
```

- a) alphabetic
- b) not alphabetic
- c) Error
- d) None of the mentioned

View Answer

Answer: a

Explanation: In this program, We are checking whether the character is alphabetic or not alphabetic by using the function isalpha.

Output:

```
$ g++ loc1.cpp
$ a.out
alphabetic
```

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: d

Explanation: In this program, We are counting the number of alphanumeric values by using the function isalnum.

Output:

```
$ g++ loc2.cpp
$ a.out
4
```

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: b

Explanation: In this program, We are counting the number of special characters in the program by using the function ispunct.

Output:

```
$ g++ loc3.cpp
$ a.out
2
```

- a) 64345
- b) 21312
- c) 65535
- d) Error

View Answer

Answer: c

Explanation: In this program, We are converting the hexadecimal number to decimal number.

Output:

```
$ g++ loc4.cpp
$ a.out
65535
```

9. What kind of locale does every program is having in C++?

- a) local locale
- b) global locale
- c) temp locale
- d) none of the mentioned

View Answer

Answer: b

Explanation: Every program has a single locale object which is its global locale.

10. What will the monetary facet will do?

- a) Handle formatting and parsing of monetary values
- b) Handle formatting and parsing of character values
- c) Parsing of character values
- d) None of the mentioned

View Answer

Answer: a

Explanation: None.

ayzom.com

C++ Programming Questions and Answers – C Input Output

1. Which header file is used with input and output operations of C in C++?

- a) `stdio.h`
- b) `cstdio`
- c) `iostream`
- d) none of the mentioned

View Answer

Answer: b

Explanation: Input and Output operations of c can be performed in C++ using the C Standard Input and Output Library.

2. Which will be used with physical devices to interact from C++ program?

- a) Programs
- b) Library
- c) Streams
- d) None of the mentioned

View Answer

Answer: c

Explanation: C++ library uses streams to operate with physical devices such as Keyboards, Printers, Terminals or with any other type of files supported by the system.

3. How many streams are automatically created when executing a program?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: c

Explanation: There are streams that are automatically created when executing a program. They are `stdin`, `stdout` and `stderr`.

- a) Error
- b) Success
- c) Runtime Error
- d) Can't say

View Answer

Answer: d

Explanation: If `myfile.txt` exists, then it will delete the file. Else it will print an error message.

Output:

```
$ g++ out.cpp
$ a.out
Success
```

- a) Count of '\$' symbol
- b) Error opening file
- c) Any of the mentioned
- d) None of the mentioned

View Answer

Answer: c

Explanation: Anyone is possible – Either the file doesn't exist or If exist, it will print the total number of '\$' character.

- a) ABCD
- b) ABC
- c) ABCDE
- d) None of the mentioned

View Answer

Answer: c

Explanation: In this program, We are printing from A to E by using the putc function.

Output:

advertisement

```
$ g++ out2.cpp
$ a.out
ABCDE
```

-
- a) name
 - b) new
 - c) newname
 - d) none of the mentioned

View Answer

Answer: c

Explanation: In this program, We are renaming the myfile2 to newname by using the function rename.

Output:

```
myfile2.txt is renamed to newname.txt
```

-
- a) 10
 - b) 15
 - c) Depends on the text file
 - d) None of the mentioned

View Answer

Answer: c

Explanation: In this program, We are reading the number of characters in the program by using the function feof.

Output:

```
$ g++ out4.cpp
$ a.out
162
```

9. How many indicators are available in c++?

- a) 4
- b) 3
- c) 2
- d) 1

View Answer

Answer: b

Explanation: There are three indicators are available in C++. They are Error indicator, End-Of-File indicator and Position indicator.

10. What is the benefit of c++ input and output over c input and output?

- a) Type safety
- b) Exception
- c) Both Type safety & Exception
- d) None of the mentioned

View Answer

Answer: a

Explanation: C++ input and output are type safety that means we don't need to specify the type of variable we are printing.

eg:

in C we need to specify %d showing that an integer will be printed, whereas in C++ we just cout the variable.

```
printf("%d", a);
```

```
cout<<a;
```

ayzom.com

C++ Programming Questions and Answers – Numeric Limits

1. To which type does the numeric limits are suitable?

- a) Character types
- b) Mixed type
- c) Arithmetic types
- d) None of the mentioned

View Answer

Answer: c

Explanation: Numeric limits provides the information about the properties of arithmetic types.

2. Where does the member should be defined if it is used in the program?

- a) Namespace scope
- b) Character scope
- c) Namespace & Character scope
- d) None of the mentioned

View Answer

Answer: a

Explanation: The member shall still be defined in a namespace scope if it is used in the program.

3. What will the max function in the numeric limit will return for type float?

- a) Maximum finite value for a float type
- b) Maximum finite value
- c) Minimum finite value
- d) None of the mentioned

View Answer

Answer: a

Explanation: Max function in the numeric limit will return the maximum finite value for a float type.

- a) 53234
- b) false
- c) true
- d) none of the mentioned

View Answer

Answer: b

Explanation: In this program, We are checking whether the integer has limit or not by using `has_infinity` function.

Output:

```
$ g++ num.cpp
$ a.out
false
```

- a) 53723
- b) 32423
- c) 32767
- d) None of the mentioned

View Answer

Answer: c

Explanation: In this program, We are finding the max range for short int by using `max` function.

Output:

```
$ g++ num1.cpp
$ a.out
32767
```

- a) 0010
- b) 0012
- c) 1100
- d) 0011

View Answer

Answer: d

Explanation: In this program, We are finding whether the types are free of rounding errors by using the function `is_exact`.

Output:

```
$ g++ num2.cpp
$ a.out
0011
```

- a) 6
- b) 15
- c) 100000000
- d) All of the mentioned

View Answer

Answer: d

Explanation: In this program, We are finding the number of decimal points that the type can represent without loss of precision.

Output:

```
$ g++ num3.cpp
$ a.out
6
15
100000000
```

- a) 125
- b) -125
- c) 123
- d) None of the mentioned

View Answer

Answer: b

Explanation: In this program, We are finding the minimum radix of a type by using `min_exponent` function.

Output:

```
$ g++ num4.cpp
$ a.out
-125
```

9. Which header file is used for the numeric limits in C++?

- a) `<iostream>`
- b) `<limits>`
- c) `<number>`
- d) None of the mentioned

View Answer

Answer: b

Explanation: `<limits>` header file will be used for numeric limits in C++.

10. Pick out the incorrect static function member in numeric limits.

- a) denorm_min
- b) digits
- c) infinity
- d) max_finite

View Answer

Answer: d

Explanation: None.

ayzom.com

C++ Programming Questions and Answers – Standard Mathematical Functions

1. With which does the trigonometric functions work with angles in c++?

- a) Degrees
- b) Radians
- c) Both Degrees & Radians
- d) None of the mentioned

View Answer

Answer: b

Explanation: The trigonometric functions work with angles in radians rather than degrees.

2. Which header file is required for manipulation of math functions in c++?

- a) cmath
- b) maths
- c) math
- d) none of the mentioned

View Answer

Answer: a

Explanation: #include is a header file required for manipulation of math functions.

3. How many macros are used by mathematical functions in the header file ?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: c

Explanation: There are three macros used in mathematical functions. They are HUGE_VAL, EDOM, ERANGE.

- a) 6
- b) 7
- c) 8
- d) 9

View Answer

Answer: c

Explanation: In this program, We are breaking out the number by using the frexp function.

Output:

```
$ g++ math.cpp
$ a.out
8
```

- a) 5.5
- b) 3.14
- c) 1.704
- d) None of the mentioned

View Answer

Answer: c

Explanation: In this program, We are finding out the log value of param by using param function.

Output:

```
$ g++ math1.cpp
$ a.out
1.704748
```

- a) 343
- b) 343.00
- c) 334
- d) None of the mentioned

View Answer

Answer: b

Explanation: In this program, We are calculating the 7 power of 3 by using the powfunction. As we are using 7.0, it is producing the result in 343.00

Output:

```
$ g++ math2.cpp
$ a.out
343.000000
```

- a) 1.300
- b) 1.700
- c) 1.300
- 1.700
- d) None of the mentioned

View Answer

Answer: c

Explanation: In this program, We are finding the remainder of the given values by using fmod function.

Output:

ayzom.com

advertisement

```
$ g++ math3.cpp
$ a.out
1.300000
1.700000
```

- a) 10
- b) -10
- c) 10.600000
- d) None of the mentioned

View Answer

Answer: c

Explanation: In this program, We are finding the absolute value of -10.6 by using abs function.

Output:

```
$ g++ math4.cpp
$ a.out
10.600000
```

9. Which of the following mathematical function is overloaded in <complex> and <valarray>?

- a) cos
- b) tan
- c) sin
- d) mod

View Answer

Answer: b

Explanation: Because tan has more different definition in normal case and in case of complex number i.e. $\tan = \sin/\cos$ (normally) and $\tan = y/x$ in complex numbers.

10. How many parameters are used in frexp function?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: b

Explanation: There are two parameters that are used in frexp function. They are floating point value to be computed and pointer to an int object.

ayzom.com

C++ Programming Questions and Answers – Vector Arithmetic

1. Which of the following library is used to do vector arithmetic?

- a) Boost
- b) Time
- c) OpenGL
- d) None of the mentioned

View Answer

Answer: a

Explanation: Boost package has a linear algebra package that may well suits for vector arithmetic.

2. Which header file is used to manipulate the vector algebra in c++?

- a) math
- b) cmath
- c) vmath
- d) none of the mentioned

View Answer

Answer: c

Explanation: vmath is set of C++ classes for Vector and Matrix algebra used in the programs.

3. What type of reference should be used in vector arithmetic?

- a) dynamic
- b) const
- c) both dynamic & const
- d) none of the mentioned

View Answer

Answer: b

Explanation: As we are using the vector and it will give accurate result if we use const reference.

a) 4, 5

b) 4, 4

c) 5, 4

d) 5, 5

View Answer

Answer: c

Explanation: In this program, We are adding two vectors by using standalone function and printing it.

Output:

```
$ g++ vecar.cpp
$ a.out
5, 4
```

a) false

b) true

c) false & true

d) none of the mentioned

View Answer

Answer: a

Explanation: In this program, We are flipping the vector values by using flip function.

Output:

advertisement

```
$ g++ vecar1.cpp  
$ a.out  
false
```

6. What will be the type of output of vector cross product?

- a) Scalar
- b) Vector
- c) Both Scalar & Vector
- d) None of the mentioned

View Answer

Answer: b

Explanation: Cross product of two vectors results into a vector.

7. Which function is used to optimize the space in vector?

- a) at
- b) bool
- c) operator
- d) none of the mentioned

View Answer

Answer: b

Explanation: This is a specialized version of vector, which is used for elements of type bool and optimizes for space.

8. What is the use of vector arithmetic in c++?

- a) Computer graphics
- b) Computer booting
- c) Both Computer graphics & Computer booting
- d) None of the mentioned

View Answer

Answer: a

Explanation: None.

C++ Programming Questions and Answers – Generalized Numeric Algorithms

1. Which header file is used to operate on numeric sequences?

- a) number
- b) numeric
- c) algorithm
- d) none of the mentioned

View Answer

Answer: b

Explanation: header file is used to operate on numeric sequences that support certain operations.

2. Which mathematics library is used for vector manipulation in c++?

- a) cli++
- b) vec++
- c) blitz++
- d) none of the mentioned

View Answer

Answer: c

Explanation: Blitz++ is a high-performance vector mathematics library written in C++.

3. What is the use of accumulate function in numeric library?

- a) Returns the number
- b) Returns the result of accumulating all the values in the range
- c) Returns the number & result
- d) None of the mentioned

View Answer

Answer: b

Explanation: Returns the result of accumulating all the values in the range from first to last.

- a) 1 1 1 2
- b) 1 2 3 1
- c) 1 2 3 5
- d) None of the mentioned

View Answer

Answer: a

Explanation: In this program, We are calculating the adjacent difference of the given range by using function adjacent_difference.

Output:

```
$ g++ gnl.cpp
$ a.out
1 1 1 2
```

- a) 100
- b) 140
- c) 160
- d) 180

View Answer

Answer: c

Explanation: In this program, We are calculating the product of every number in the given range by using accumulate function.

Output:

```
$ g++ gn11.cpp
$ a.out
160
```

- a) 1 3 6
- b) 1 3 6 10 15
- c) 1 3 6 10 16
- d) None of the mentioned

View Answer

Answer: b

Explanation: In this program, We are calculating the sum of the given range by using `partial_sum` function.

Output:

```
$ g++ gn12.cpp
$ a.out
1 3 6 10 15
```

- a) 40
- b) 100
- c) 140
- d) None of the mentioned

View Answer

Answer: a

Explanation: In this program, We are finding the difference between the init and the total of numbers range.

Output:

```
$ g++ gn13.cpp
$ a.out
40
```

- a) 40
 - b) 34
 - c) 32
 - d) 20
- View Answer

Answer: b

Explanation: In this program, We are forming the custom function from two ranges by using `inner_product` function.

Output:

```
$ g++ gn14.cpp
$ a.out
34
```

9. How many parameters are available in `partial_sum` function in c++?

- a) 2
 - b) 3
 - c) 2 or 3
 - d) 3 or 4
- View Answer

Answer: d

Explanation: There are three or four parameters available in `partial_sum` function in C++. They are first and last element, result and an optional binary operator.

10. What is the default operation of adjacent_difference function in numeric library?

- a) Difference
- b) Addition
- c) Multiplication
- d) None of the mentioned

View Answer

Answer: a

Explanation: The default operation is to calculate the difference, but some other operation can be specified as binary operator instead.

ayzom.com

C++ Programming Questions and Answers – Random Numbers

1. Which header file is used to create the pseudo random generator?

- a) random
- b) cstdlib
- c) rand
- d) both random and cstdlib

View Answer

Answer: d

Explanation: cstdlib header file is used to create pseudo random number. C++11 standard added random header file as well to generate random numbers.

2. Which is a constant defined in <cstdlib> header file?

- a) RAND_MAX
- b) Rand
- c) Srand
- d) None of the mentioned

View Answer

Answer: a

Explanation: RAND_MAX is a constant defined in <cstdlib> for deciding the maximum random number that can be produced.

3. How many parameters are available in srand function?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: a

Explanation: There is one parameter available in srand function. That is an integer value to be used as seed by the pseudo-random number generator algorithm.

- a) Any number
- b) 89
- c) 0 to RAND_MAX
- d) None of the mentioned

View Answer

Answer: c

Explanation: As the declared number is integer, It will produce the random number from 0 to RAND_MAX. The value of RAND_MAX is library-dependent, but is guaranteed to be at least 32767 on any standard library implementation.

Output:

```
$ g++ rand.cpp
$ a.out
574
```

- a) 4385234
- b) 12321412
- c) Depends on the compiler
- d) None of the mentioned

View Answer

Answer: c

Explanation: RAND_MAX is a function used by the compiler to create a maximum random number.

Output:

```
$ g++ rand1.cpp
$ a.out
2147483647
```

- a) 12
- b) 23
- c) 33
- d) Any number from 0 to 99

View Answer

Answer: d

Explanation: This program will create a random number based on time function using srand function.

Output:

```
$ g++ rand2.cpp
$ a.out
23
```

- a) 2 4
- b) 10 20
- c) Any two number from 1 to 10
- d) None of the mentioned

View Answer

Answer: c

Explanation: In this program, It will produce two numbers from 1 to 10 by using srand function.

Output:

advertisement

```
$ g++ rand3.cpp
$ a.out
4 5
```

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: a

Explanation: In this program the output will always be 1 because as one can observe from the formula used in line #13 that the term inside int() will always be zero. Hence the output will always be 1.

Output:

```
$ g++ rand4.cpp
$ a.out
1
```

9. Which operator is used to produce a certain number in a specific range?

- a) \$
- b) %
- c) modulo operator
- d) both % and modulo operator

View Answer

Answer: d

Explanation: In C++, modulo operator is denoted by %.

10. Which can be used to create a random number without duplication?

- a) Character
- b) Time
- c) Both Character & Time
- d) None of the mentioned

View Answer

Answer: b

Explanation: None.

ayzom.com

C++ Programming Questions and Answers – File Handling

1. Which header file is required to use file I/O operations?

- a) <ifstream>
- b) <ostream>
- c) <fstream>
- d) <iostream>

View Answer

Answer: c

Explanation: <fstream> header file is needed to use file I/O operations in C++. This header file contains all the file I/O operations definition.

2. Which of the following is used to create an output stream?

- a) ofstream
- b) ifstream
- c) iostream
- d) fstream

View Answer

Answer: a

Explanation: ofstream is used to create an output stream in C++ file handling operations. Ofstream objects are used to read files.

3. Which of the following is used to create a stream that performs both input and output operations?

- a) ofstream
- b) ifstream
- c) iostream
- d) fstream

View Answer

Answer: d

Explanation: fstream is used to create a stream that performs both input and output operations in C++ file handling.

4. Which of the following is not used as a file opening mode?

- a) ios::trunc
- b) ios::binary
- c) ios::in
- d) ios::ate

View Answer

Answer: a

Explanation: ios::trunc is used to truncate a file if it exists. It is not a file opening mode.

5. Which of the following statements are correct?

- 1) It is not possible to combine two or more file opening mode in open() method.
 - 2) It is possible to combine two or more file opening mode in open() method.
 - 3) ios::in and ios::out are input and output file opening mode respectively.
- a) 1, 3
 - b) 2, 3
 - c) 3 only
 - d) 1, 2

View Answer

Answer: a

Explanation: C++ allows to use one or more file opening mode in a single open() method. ios::in and ios::out are input and output file opening mode respectively.

6. By default, all the files in C++ are opened in _____ mode.

- a) Text
 - b) Binary
 - c) ISCII
 - d) VTC
- View Answer

Answer: a

Explanation: By default, all the files in C++ are opened in text mode. They read the file as normal text.

7. What is the use of ios::trunc mode?

- a) To open a file in input mode
 - b) To open a file in output mode
 - c) To truncate an existing file to half
 - d) To truncate an existing file to zero
- View Answer

Answer: d

Explanation: In C++ file handling, ios::trunc mode is used to truncate an existing file to zero length.

8. Which of the following is the default mode of the opening using the ofstream class?

- a) ios::in
 - b) ios::out
 - c) ios::app
 - d) ios::trunc
- View Answer

Answer: b

Explanation: By default, the file is opened in ios::out mode if the file object we are using is of ofstream class.

9. What is the return type open() method?

- a) int
 - b) char
 - c) bool
 - d) float
- View Answer

Answer: c

Explanation: open() method returns a bool value indicating whether the file is opened or some error has occurred.

10. Which of the following is not used to seek file pointer?

- a) ios::set
 - b) ios::end
 - c) ios::cur
 - d) ios::beg
- View Answer

Answer: a

Explanation: ios::set is not used to seek file pointer. ios::end is used to seek from the end of the file. ios::curr from the current position. ios::beg from the beginning.

11. Which of the following is the default mode of the opening using the ifstream class?

- a) ios::in
 - b) ios::out
 - c) ios::app
 - d) ios::trunc
- View Answer

Answer: a

Explanation: By default, the file is opened in ios::in mode if the file object we are using is of ifstream class.

12. Which of the following is the default mode of the opening using the fstream class?

- a) ios::in
- b) ios::out
- c) ios::in|ios::out
- d) ios::trunc

View Answer

Answer: c

Explanation: By default, the file is opened in ios::in|ios::out mode if the file object we are using is of fstream class.

13. Which function is used in C++ to get the current position of file pointer in a file?

- a) tell_p()
- b) get_pos()
- c) get_p()
- d) tell_pos()

View Answer

Answer: a

Explanation: C++ provides tell_p() function to get the current position of the file pointer in a file.

14. Which function is used to reposition the file pointer?

- a) moveg()
- b) seekg()
- c) changep()
- d) go_p()

View Answer

Answer: b

Explanation: seekg() function is used to reposition a file pointer in a file. The function takes the offset and relative position from where we need to shift out pointer.

15. Which of the following is used to move the file pointer to start of a file?

- a) ios::beg
- b) ios::start
- c) ios::cur
- d) ios::first

View Answer

Answer: a

Explanation: ios::beg is used to reposition the file pointer to the beginning of the file. It is whenever you want to reposition the pointer at the beginning from any point to the start of the file.

C++ Programming Questions and Answers – Lambda Expressions

1. What is lambda expression in C++?

- a) A technique of C++ that allows us to write inline functions without a name
- b) A technique of C++ that allows us to write overloaded functions
- c) A technique of C++ that allows us to write functions that are called more than once
- d) A technique of C++ that allows us to write functions without parameters

View Answer

Answer: a

Explanation: Lambda expression is a technique available in C++ that helps the programmer to write inline functions that will be used once in a program and so there is no need of providing names to them. Hence they are a type of inline functions without names.

2. What is the syntax of defining lambda expression?

- a) [capture clause](parameters) -> return_type { body of the function }
- b) [parameters](capture clause) -> return_type { body of the function }
- c) [parameters:capture clause]() -> return_type { body of the function }
- d) [capture clause:parameters]() -> return_type { body of the function }

View Answer

Answer: a

Explanation: The correct syntax of defining a lambda expression is given below:

```
[capture clause](parameters) -> return_type  
{  
    the body of the function  
}
```

3. What is the correct statement about lambda expression?

- a) The return type of lambda expression can be neglected in some cases
- b) The return type of lambda expression must be specified in all cases
- c) Lambda expression should be very large functions
- d) Lambda expression is also available in C

View Answer

Answer: a

Explanation: Return type in lambda expression can be ignored in some cases as the compiler will itself figure that out but not in all cases. Lambda expression is used to define small functions, not large functions. Lambda expression is introduced in C++.

4. In how many ways we can capture the external variables in the lambda expression?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: c

Explanation: There are three ways in which we can capture the external variables inside the lambda expression namely capture by reference, capture by value and capture by both that is mixed capture.

5. Which of the following operator is used to capture all the external variable by reference?

- a) &
- b) =
- c) *
- d) &&

View Answer

Answer: a

Explanation: The lambda expression uses & operator to capture the external variable by reference.

6. Which of the following operator is used to capture all the external variable by value?

- a) &
- b) =
- c) *
- d) &&

View Answer

Answer: b

Explanation: The lambda expression uses = operator to capture the external variable by value.

7. Which is the correct syntax of capturing a variable 'X' by reference and other variable 'Y' by value in lambda expression?

- a) [&X, Y]
- b) [X, &y]
- c) [X, Y]
- d) [&x, &Y]

View Answer

Answer: a

Explanation: In order to capture a variable by reference we use & operator whereas when we capture a single variable by value then we just write the name of that variable without any operator preceding it, So the correct way of capturing the variables X and Y, in this case, is [&X, Y].

- a) 1
- b) 0
- c) Error
- d) Segmentation fault

View Answer

Answer: c

Explanation: The above code gives an error because x is neither passed as a parameter in lambda expression nor it is declared as a local variable inside the expression. So the only x that will be referred is the outside x but as the lambda expression does not capture any variable, therefore, it is also not allowed to access the external variable x hence as variable x is not defined therefore the program gives the error.

- a) 0
- b) 1
- c) Error
- d) Segmentation fault

View Answer

Answer: b

Explanation: The program is correct. In this program you can observe that we have specified the return type of the expression though also the program runs fine because compiler is able to find out the return type of the expression.

- a) Value of a: 5
- b) Value of a: 10
- c) Error
- d) Segmentation fault

View Answer

Answer: c

Explanation: As this lambda expression is capturing the external variable by value therefore the value of a cannot be changes inside the lambda expression hence the program gives error.

- a) Value of a: 5
- b) Value of a: 10
- c) Error
- d) Segmentation fault

View Answer

Answer: b

Explanation: As this lambda expression is capturing the external variable by reference therefore the change in value of a will be reflected back outside the expression therefore the value of a will now be 10 and 10 is printed.

- a) Value of a: 5
- b) Value of a: 10
- c) Error
- d) Segmentation fault

View Answer

Answer: c

Explanation: As this lambda expression is not capturing variable b but trying to access the external variable b hence the program gives an error.

ayzom.com

C++ Programming Questions and Answers – Command Line Arguments

1. What are command line arguments?

- a) Arguments passed to main() function
- b) Arguments passed to any function
- c) Arguments passed to class functions
- d) Arguments passed to structure functions

View Answer

Answer: a

Explanation: Command line arguments are the arguments that passed to the main function when the program is starting its execution.

2. To use command line arguments in C++, how many parameters are passed to the main function?

- a) 1
- b) 2
- c) 3
- d) 4

View Answer

Answer: b

Explanation: 2 arguments are needed to be passed to main() function while using command line arguments. The first one represents a number of strings in the argument list and the second list represents the list of string arguments.

3. What is the signature of main in function using command line arguments?

- a) `int main(int argc, char const *argv[]);`
- b) `int main(int argc, char const **argv);`
- c) `int main(int argc, char **argv);`
- d) all of the mentioned

View Answer

Answer: d

Explanation: Any of the above signature can be used while using command line arguments in C++ programs.

4. What does the first parameter of the main function represent?

- a) Number of command line arguments
- b) List of command line arguments
- c) Dictionary of command line arguments
- d) Stack of command line arguments

View Answer

Answer: a

Explanation: The first argument of the main() function represents the number of command line arguments that are passed.

5. What does the second parameter of the main function represent?

- a) Number of command line arguments
- b) List of command line arguments
- c) Dictionary of command line arguments
- d) Stack of command line arguments

View Answer

Answer: b

Explanation: The second argument of the main() function represents the list of command line arguments that are passed.

6. Which of the following is correct about the first parameter of the main function?

- a) First argument is of int type

- b) Stores the count of command line arguments
- c) First argument is non-negative
- d) All of the mentioned

View Answer

Answer: d

Explanation: All of the statements about the first parameter is correct. The first parameter is of non-negative integer type and stores the count of command line arguments.

7. Which of the following is correct about the second parameter of the main function?

- a) Second parameter is an array of character pointers
- b) First string of the list is the name of the program's output file
- c) The string in the list are separated by space in the terminal
- d) All of the mentioned

View Answer

Answer: d

Explanation: All of the statements about the second parameter is correct. It is the collection of character pointers which of which the first represents the name of the program file.

8. Which of the following gives the name of the program if the second parameter to the main function is `char **argv`?

- a) `argv[3]`
- b) `argv[1]`
- c) `argv[0]`
- d) `argv[2]`

View Answer

Answer: c

Explanation: The first string in the list of command line arguments represents the name of the program which can be accessed by using `argv[0]`.

View Answer

Answer: a

Explanation: In this program we are trying to print all the command line arguments. Hence as the list contains `["./output", "Hello", "World"]` so the output is as shown. The first string is not `program.cpp` because the first string represents the name of the output file of the program.

10. Which character is used to separate different arguments?

- a) `#`
- b) `$`
- c) space
- d) `|`

View Answer

Answer: c

Explanation: Command line arguments are separated by space. So if you write

`./output This is a single parameter`

then they will be interpreted as 5 command line arguments as shown: `["./output", "This", "is", "single", "parameter"]`.

11. Which is the correct way of handling arguments with spaces?

- a) Use single quotes
- b) Either single or double quotes
- c) Use double quotes
- d) There is no way of handling arguments with space

View Answer

Answer: b

Explanation: One can use either single or double quotes to handle command line argument with spaces in-between. For example, `./output "Hello World"` has 2 command line arguments: `./output` and `"Hello World"`.

12. Which of the following is correct to interpret Hello World as a single argument?

1) \$./output 'Hello World'

2) \$./output Hello World

a) Only 1

b) Only 2

c) Both 1 and 2

d) Neither 1 nor 2

View Answer

Answer: c

Explanation: To interpret space separated words as single argument one can use single or double quotes.

ayzom.com